

LAPORAN MAGANG
PT. ASTRA SEDAYA FINANCE



Dipersiapkan oleh:

Valentino Budiono

150708614

Program Studi Teknik Informatika
Fakultas Teknologi Industri
Universitas Atma Jaya Yogyakarta
2019

HALAMAN PENGESAHAN

Proposal Magang

Proposal ini telah diperiksa dan disetujui

Pada tanggal 21 Mei 2019

Oleh :

Dosen Pembimbing



Findra Kartikasari D, ST, M.M, M.T.

Pembimbing Lapangan



Julia Christie Feliciano D. C.

KATA PENGANTAR

Puji Syukur dan terima kasih penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan laporan magang ini. Penyusunan laporan magang ini tidak lepas dari bantuan berbagai pihak yang senantiasa mendukung penulis. Oleh sebab itu penulis memberikan ucapan terima kasih kepada :

1. Tuhan Yang Maha Esa, yang selalu menyertai dan melimpahkan rahmat dan karunia-Nya kepada penulis.
2. Orang tua, yang selalu mendukung penulis.
3. Ibu Findra Kartikasari Dewi, ST, M.M, M.T. selaku dosen pembimbing magang yang selama ini telah membimbing penulis selama magang dan penulisan laporan ini.
4. Ibu Awa dan Ibu Christie, selaku pembimbing lapangan yang telah membimbing penulis selama pelaksanaan magang.
5. Teman-teman Gala (Desya, Dwi, Glady, Mike, Cherlie, Felix, Eric, Alan, Krisna) yang telah memberikan semangat kepada penulis untuk melaksanakan dan menyelesaikan magang.
6. Teman-teman kos Maranatha yang selalu mendukung penulis.
7. Seluruh teman-teman dan saudara yang telah memberikan bantuan serta dukungan pada penulis.

Penyusunan laporan ini bertujuan untuk memenuhi persyaratan dalam menempuh mata kuliah magang. Laporan ini berisi dokumentasi pengetahuan, pengalaman, hingga hambatan yang penulis dapatkan selama melaksanakan magang. Magang dilaksanakan penulis pada tanggal 03 Januari 2019 - 07 April 2019 di PT. Astra Sedaya Finance (ACC – Astra Credit Company).

Penulis menyadari bahwa pelaksanaan magang dan penyusunan laporan ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan. Penulis berharap laporan ini akan memberikan manfaat kepada pembaca.

Yogyakarta, 1 Mei 2019



Penulis,
Valentino Budiono

DAFTAR ISI

HALAMAN PENGESAHAN.....	ii
KATA PENGANTAR.....	iii
DAFTAR GAMBAR.....	vi
BAB I PENDAHULUAN.....	1
1.1 Sekilas Astra Credit Company	1
1.2 Sejarah Singkat Astra Credit Company	2
1.3 Visi dan Misi Astra Credit Company	4
1.4 Struktur Organisasi Astra Credit Company	4
1.5 Deskripsi Tugas dari Struktur Organisasi Astra Credit Company	5
1.6 Departemen IT dalam Perusahaan	11
BAB II PELAKSANAAN MAGANG	12
2.1 Penjelasan Logbook	12
2.2 Hasil Pekerjaan Secara Umum	36
2.3 Bukti Hasil Pekerjaan	36
BAB III HASIL PEMBELAJARAN	62
3.1 Manfaat Magang.....	62
3.2 Penerapan Ilmu dalam Magang.....	63
BAB IV KESIMPULAN	64

DAFTAR GAMBAR

Gambar 1.1 Logo perusahaan Astra Sedaya Finance (ACC)	1
Gambar 1.2 Bagan struktur organisasi ACC	4
Gambar 2.1 Aplikasi OutSystem	36
Gambar 2.2 Aplikasi uniPaaS.....	37
Gambar 2.3 Aplikasi Software AG Designer.....	37
Gambar 2.4 Welcome Page Laravel	38
Gambar 2.5 FortiClient VPN.....	38
Gambar 2.6 Flowchart PL/SQL yang dibuat	39
Gambar 2.7 Website AccWhistle	39
Gambar 2.8 Logo Ionic	40
Gambar 2.9 API Login	40
Gambar 2.10 API Register	40
Gambar 2.11 API getAllCategory	41
Gambar 2.12 API getAllFAQ	41
Gambar 2.13 API getAllSubCategory	42
Gambar 2.14 Fungsi getMyTicketList	42
Gambar 2.15 Halaman FAQ pada Aplikasi Android	43
Gambar 2.16 Halaman Profile setelah Login	44
Gambar 2.17 Halaman Lacak Tiket	45

Gambar 2.18 Halaman Reset Password	46
Gambar 2.19 Email untuk Reset Password	47
Gambar 2.20 Halaman Home	48
Gambar 2.21 Tag Khusus dari Ionic	49
Gambar 2.22 Kode untuk filterisasi tipe file	49
Gambar 2.23 Halaman Lapor setelah Login dengan auto fill email	50
Gambar 2.24 Halaman Lapor saat Pengguna belum Login	51
Gambar 2.25 Kode untuk filterisasi file diatas 10 MB	52
Gambar 2.26 Kode ngFor dari AngularJS pada file HTML	52
Gambar 2.27 Bug yang ditemukan dilist pada google spreadsheets	52
Gambar 2.28 Bug Pertama	53
Gambar 2.29 Bug Kedua	54
Gambar 2.30 Request QC Pertama	54
Gambar 2.31 Request QC Kedua	55
Gambar 2.32 Login Screen Baru	56
Gambar 2.33 Bug Ketiga	57
Gambar 2.34 Bug Keempat	58
Gambar 2.35 Request QC Ketiga	59
Gambar 2.36 Request QC Keempat	60
Gambar 2.37 Bug Kelima	60
Gambar 2.38 Bug Kelima	61

BAB I

PENDAHULUAN

1.1 Sekilas Astra Credit Company

Astra Credit Companies adalah grup perusahaan pembiayaan yang telah berdiri sejak tahun 1982. Pada tahun 1994, PT Astra Sedaya Finance bersama PT Swadharma Bhakti Sedaya Finance (SBSF), PT Astra Auto Finance (AAF), PT Staco Estika Sedaya Finance (SESF) dan PT Pratama Sedaya Finance (PSF) tergabung dalam kelompok pembiayaan dengan brand Astra Credit Companies (ACC). ACC memiliki beragam produk pembiayaan, mulai dari pembiayaan mobil baru untuk produk Astra dan non Astra, pembiayaan mobil bekas melalui dealer maupun perorangan dan juga melayani pembiayaan Retail Commercial dan Fleet. Untuk logo perusahaan ini sendiri dapat dilihat pada Gambar 1.1.



Gambar 1.1 Logo perusahaan Astra Credit Company

ACC memiliki jaringan yang luas meliputi 73 kantor cabang dan kantor pelayanan. Dengan tersedianya lebih dari 76.000 titik pembayaran di seluruh Indonesia akan membuat pelanggan ACC lebih mudah melakukan pembiayaan di ACC. ACC sebagai perusahaan multifinance yang solid dan telah menerima pengakuan berupa penghargaan dari berbagai pihak secara kontinyu. Di tahun 2015 ini ACC telah menerima penghargaan Top Brand

Award, Contact Center Service Excellence Award, Service Quality Award dan Corporate Image Award. ACC juga mendapatkan penghargaan atas partisipasi dalam Operasionalisasi mobil SiMOLEK pada program Literasi Keuangan Otoritas Jasa Keuangan.

1.2 Sejarah Singkat Astra Credit Company

Astra Credit Companies atau biasa di singkat dengan **ACC** adalah perusahaan pembiayaan mobil dan alat berat. Sesuai dengan Peraturan Otoritas Jasa Keuangan No. 29/POJK.05/2014 ACC melakukan perluasan usaha di bidang Pembiayaan Investasi, Pembiayaan Modal Kerja, Pembiayaan Multiguna dan Sewa Operasi (Operating Lease), baik dengan skema konvensional maupun syariah. PT Astra Sedaya Finance yang merupakan cikal bakal ACC berdiri pada 15 Juli 1982 dengan nama PT Rahardja Sedaya, didirikan guna mendukung bisnis otomotif kelompok Astra.

Di tahun 1990, PT Rahardja Sedaya berganti nama menjadi PT Astra Sedaya Finance. Dalam perkembangannya, PT Astra Sedaya Finance memiliki penyertaan saham pada perusahaan asosiasi, yaitu PT Swadharma Bhakti Sedaya Finance, PT Pratama Sedaya Finance, PT Staco Estika Sedaya Finance dan PT Astra Auto Finance yang semuanya telah terdaftar dan diawasi oleh Otoritas Jasa Keuangan (OJK) serta mendapatkan izin dari Departemen Keuangan Republik Indonesia Direktorat Jenderal Lembaga Keuangan dengan nomor izin usaha sebagai berikut :

- PT Astra Sedaya Finance dengan nomor izin usaha 1093/KMK.013/1989
- PT Swadharma Bhakti Sedaya Finance dengan nomor izin usaha 1095/KMK.013/1989
- PT Astra Auto Finance dengan nomor izin usaha 437/KMK.017/1995

- PT Staco Estika Sedaya Finance dengan nomor izin usaha 590/KMK.013/1990
- PT Pratama Sedaya Finance dengan nomor izin usaha 1258/KMK.013/1989

Sejak tahun 1994, PT Astra Sedaya Finance dan perusahaan asosiasinya mengembangkan merek Astra Credit Companies untuk mendukung usahanya. ACC berkomitmen penuh untuk meningkatkan layanan pada masyarakat. ACC menyediakan fasilitas pembiayaan untuk pembelian mobil dan alat berat dalam kondisi baru ataupun bekas serta fasilitas Pembiayaan Investasi, Pembiayaan Modal Kerja, Pembiayaan Multiguna dan Sewa Operasi (Operating Lease). ACC juga mendukung penjualan mobil melalui jaringan dealer, showroom maupun perseorangan di seluruh wilayah Indonesia. Jaringan ACC tersebar di hampir seluruh kota besar di Indonesia. Saat ini ACC memiliki 75 kantor cabang yang tersebar di 59 kota di Indonesia, dan akan terus bertambah.

ACC selalu mempertahankan reputasinya sebagai perusahaan pembiayaan terkemuka di Indonesia. Saat krisis ekonomi melanda Indonesia di tahun 1998, ACC dapat melewati krisis ini dengan baik. Termasuk juga dapat melunasi pinjaman sindikasi pada tahun 1999, tanpa restrukturisasi.

Pada tahun 2000, ACC mulai melakukan penerbitan obligasi dengan rating A- dari PT Pemeringkat Efek Indonesia. Saat ini ACC telah meraih peringkat AAA (idn) Stabel Outlook dari PT Fitch Ratings Indonesia dan peringkat idAAA dari PT Pemeringkat Efek Indonesia. ACC sebagai perusahaan pembiayaan terbesar dan terkemuka di Indonesia, selalu membayarkan nilai pokok hutang dan bunga tepat pada waktunya.

1.3 Visi dan Misi Astra Credit Company

Visi

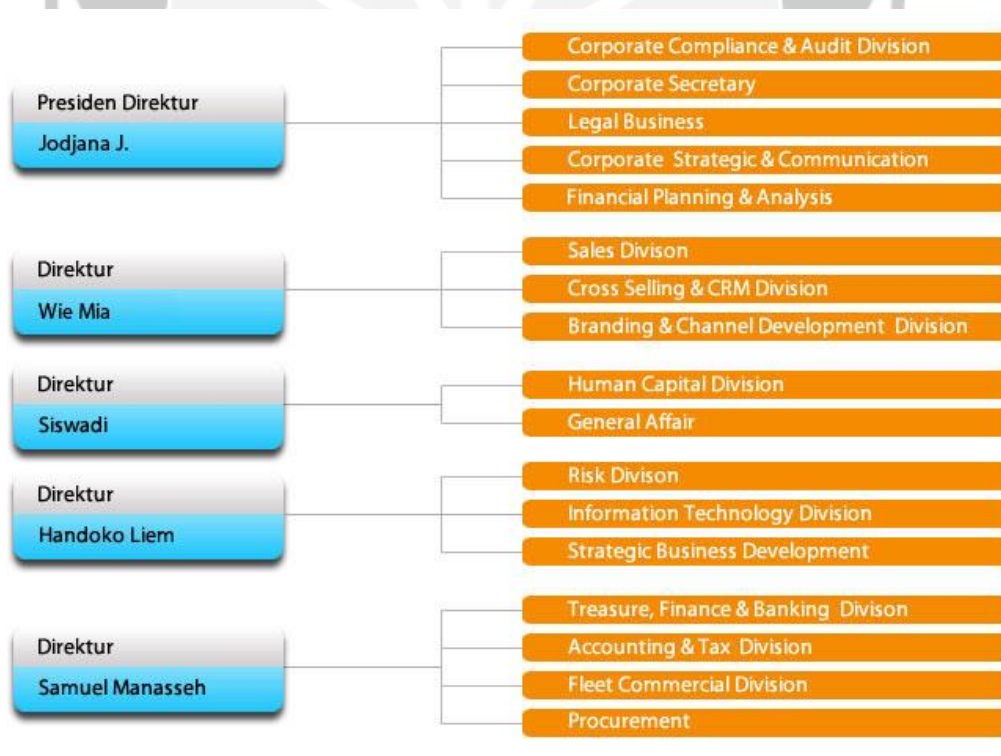
Become the 1st Choice Financing Company with Total Solution.

Misi

To Promote Credit for a better living.

1.4 Struktur Organisasi Astra Credit Company

Struktur organisasi pada perusahaan ACC, terdapat 1 Presiden Direktur bernama Jodjana J. dan terdapat 4 Direktur diantaranya Wie Mia, Siswadi, Handoko Liem dan Samuel Manasseh, setiap Direktur disini membawahi beberapa divisi seperti yang terlihat pada Gambar 1.2.



Gambar 1.2 Struktur Organisasi Perusahaan Astra Credit Company

1.5 Deskripsi Tugas dari Struktur Organisasi Astra Credit Company

1. Direksi

- a. Menyusun visi, misi dan nilai-nilai serta rencana strategis Perseroan dalam bentuk rencana korporasi (*corporate plan*) dan rencana bisnis (*business plan*).
- b. Menetapkan struktur organisasi Perseroan lengkap dengan rincian tugas setiap divisi dan unit usaha.
- c. Mengelola sumber daya yang dimiliki Perseroan secara efektif dan efisien.
- d. Membentuk sistem pengendalian internal Perseroan dan manajemen risiko.
- e. Memperhatikan kepentingan yang wajar dari pemangku kepentingan Perseroan.

2. Corporate Audit & Compliance

- a. Meninjau kredibilitas dan obyektivitas laporan keuangan.
- b. Meninjau tindakan yang telah diambil untuk mengidentifikasi dan mengendalikan risiko bisnis dan keuangan.
- c. Meninjau tindakan yang telah diambil untuk memastikan efektivitas pengendalian internal.
- d. Meninjau rencana dan pelaksanaan kegiatan fungsi Auditor Internal, Manajemen Risiko, dan Auditor Eksternal untuk menilai apakah risiko-risiko kunci telah dengan tepat dievaluasi dan dikendalikan.
- e. Meninjau obyektivitas dan independensi Auditor Internal dan Eksternal.
- f. Meninjau kecukupan monitoring ketaatan terhadap aturan dan etika bisnis.

3. Corporate Secretary

- a. Mengikuti perkembangan Pasar Modal khususnya peraturan-peraturan yang berlaku di Pasar Modal.
- b. Memberikan pelayanan informasi kepada masyarakat yang berkaitan dengan kondisi Perseroan.
- c. Memberikan masukan kepada Direksi dalam rangka mematuhi ketentuan dan peraturan di Pasar Modal.
- d. Menjadi penghubung antara Perseroan dengan otoritas dan regulator Pasar Modal serta penghubung antara Perseroan dengan masyarakat.

4. Legal Business

- a. Menangani dokumen dan perizinan.
- b. Menangani permasalahan hukum, baik untuk masalah perdata maupun pidana.
- c. Melakukan penyesuaian-penyesuaian terhadap peraturan baru yang dikeluarkan oleh pemerintah yang berkaitan dengan operasional perusahaan.

5. Corporate Strategic & Communication

- a. Untuk menyempurnakan profil dari perusahaan (*corporate branding*).
- b. Untuk meminimalkan perbedaan antara identitas perusahaan yang diinginkan dan fitur merek dari sebuah produk yang dihasilkan perusahaan.
- c. Untuk mendelegasikan tugas-tugas dalam komunikasi.
- d. Untuk merumuskan dan melaksanakan prosedur yang efektif untuk membuat keputusan mengenai urusan komunikasi.
- e. Untuk menggerakkan dukungan internal dan eksternal demi tujuan-tujuan perusahaan.

- f. Untuk berkoordinasi dengan perusahaan-perusahaan bisnis internasional.

6. Financial Planning & Analysis

- a. Mengatur dan mengelola keuangan secara tepat.
- b. Memantau rencana keuangan.
- c. Memberikan solusi terhadap masalah keuangan yang dihadapi.

7. Sales Division

- a. Aktif mencari target.
- b. Merekap data hasil penjualan.
- c. Menjamin kepuasan pelanggan.
- d. Mencari mitra kerja.
- e. Menyusun strategi lanjutan.

8. Cross Selling & CRM Division

- a. Meningkatkan hubungan antara perusahaan dengan pelanggan yang sudah ada untuk meningkatkan pendapatan perusahaan.
- b. Menyediakan informasi yang lengkap mengenai pelanggan untuk memaksimalkan jalinan hubungan pelanggan dengan perusahaan melalui penjualan secara *up selling* dan *cross selling*.
- c. Menggunakan informasi yang terintegrasi untuk menghasilkan pelayanan yang memuaskan dengan memanfaatkan informasi pelanggan untuk memenuhi kebutuhan pelanggan.
- d. Menghasilkan konsistensi dalam prosedur dan proses menyalurkan jawaban kepada pelanggan.

9. Branding & Channel Development Division

- a. Menjalankan evaluasi berdasarkan sistem & kriteria yang telah ditetapkan guna mendapatkan profit baik dari aspek kerja dan kompetensinya sebagai dasar program pengembangan.

10. Human Capital Division

- a. Memimpin dan melaksanakan proses *recruitment*.
- b. Mengelola pelatihan & pengembangan.

11. General Affair

- a. Melakukan pengurusan seluruh perijinan yg dibutuhkan oleh perusahaan, menjaga hubungan baik dengan lingkungan sekitar perusahaan dan dengan pemerintah daerah setempat, melakukan pencatatan dan pelaporan Asset Perusahaan, melakukan pemeliharaan dan perbaikan sarana kantor, dan memastikan ketersediaan kebutuhan kantor, serta pengamanan fasilitas kantor dan asset perusahaan.

12. Risk Division

- a. Memantau dan koreksi tindakan penyimpangan.
- b. Mengawasi manajemen dan kultur pengendalian.
- c. Identifikasi penilaian risiko.

13. Information Technology Division

- a. Menilai inovasi teknologi baru dengan melakukan perbandingan (*benchmarking*), ujicoba dan analisis untuk mendapatkan teknologi yang sesuai dengan proses bisnis perusahaan.

- b. Menganalisis sistem dan teknologi yang ada di perusahaan dengan cara *outsourcing* agar dapat mengetahui sistem dan teknologi yang aplikatif.
- c. Membuat usulan dan saran-saran penyesuaian program dan strategi dibidang Sistem dan Teknologi Informasi yang dipandang perlu dengan cara menyampaikan hasil evaluasi dalam rangka pencapaian tujuan jangka panjang.
- d. Merencanakan manajemen sistem informasi dengan membuat strategi, metodologi sistem informasi agar dapat diaplikasikan.
- e. Mengelola penerapan teknologi baru menggunakan prinsip-prinsip manajemen perubahan dengan cara sosialisasi yang terencana untuk meminimalkan resistensi pengguna.

14. Strategic Business Development

- a. Menciptakan ide.
- b. Melakukan seleksi ide.
- c. Mengembangkan konsep.
- d. Melakukan sample atau uji coba.
- e. Melakukan evaluasi program.
- f. Mempelajari strategi pesaing.
- g. Mengawasi program yang telah berjalan.

15. Treasure Finance & Banking Division

- a. Mengambil sebagian atau sebuah data yang diberikan oleh bagian akunting ke dalam sebuah sistem secara komputerisasi untuk dilakukan *cash forecasting*.
- b. Mengelola modal kerja.
- c. Mengelola kas agar dana terjamin mencukupi kebutuhan kegiatan operasional sebuah perusahaan.

16. Accounting & Tax Division

- a. Mencatat, memeriksa, dan melaporkan semua transaksi yang berkaitan dengan keuangan kantor.
- b. Menyusun rencana perpajakan untuk optimalisasi pajak.
- c. Melakukan koordinasi dengan perusahaan afiliasi dan bagian terkait dalam melaksanakan hak dan kewajiban perpajakan.
- d. Approval laporan pajak masa dan tahunan secara akurat dan tepat waktu.
- e. Melakukan verifikasi transaksi perusahaan yang terkait aspek pajak.
- f. Menangani audit pajak dan menyusun budget tahunan bagian pajak.
- g. *Update* peraturan perpajakan untuk memastikan *tax compliance*.

17. Fleet Commercial Division

- a. Menghilangkan atau meminimalkan risiko yang terkait dengan investasi kendaraan.
- b. Meningkatkan efisiensi, produktivitas.
- c. Mengurangi transportasi secara keseluruhan dan staf biaya.

18. Procurement

- a. Perencanaan pembelian.
- b. Membuat prosedur standar pengadaan barang/jasa.
- c. Membuat spesifikasi barang/jasa yang dibutuhkan secara detail dengan informasi yang didapat dari departemen/bagian lain yang memintanya.
- d. Pencarian *supplier/vendor* yang tepat dengan melihat penawaran serta rekam jejaknya secara detail.

- e. Membuat perbandingan biaya pembelian dari *supplier/vendor*.
- f. Negosiasi harga, jangka waktu pembayaran (*term of payment*), pengiriman (*shipping*), dll.
- g. Memutuskan pembelian dari *supplier/vendor*.
- h. Membuat kontrak.
- i. Melakukan kontrol jumlah persediaan di gudang.
- j. Menerima tagihan pembayaran dari *vendor/supplier*.

1.6 Departemen IT dalam Perusahaan

Departemen IT atau dalam Astra Credit Company menyebutnya *Information Technology Division*. Divisi ini merupakan bagian yang berhubungan dengan semua yang berkaitan dengan IT, seperti membangun sistem baru yang digunakan dalam perusahaan tersebut, selain itu juga menganalisis sistem yang ada dan membuat usulan serta saran-saran penyesuaian program yang kemudian digunakan untuk menyempurnakan sistem yang ada. Selain itu, tugas lainnya adalah menilai inovasi teknologi terbaru dengan melakukan perbandingan, uji coba dan analisis untuk mendapatkan teknologi yang sesuai dengan proses bisnis perusahaan.

BAB II

PELAKSANAAN MAGANG

2.1 Penjelasan Logbook

Kamis, 3 Januari 2019

Hari ini, saya dan empat rekan saya (Elen, Kevin, Desya, dan Budi) selaku mahasiswa *Internship Technocenter ACC Batch 2* berkenalan dengan pembimbing lapangan kami, yaitu ibu Awa. Ibu Awa disini adalah BSA (*Business System Analyst*) dari ACC Jakarta yang akan membimbing kami hingga Februari 2019. Pada hari pertama ini kami diperkenalkan secara singkat mengenai perusahaan ACC. Pada minggu pertama kami mendapatkan pelatihan terlebih dahulu. Pelatihan pertama adalah pelatihan tentang arsitektur, pelatihan ini dibawakan oleh bapak Amri. Setelah itu ada pelatihan mengenai STLC yang dibawakan oleh Ibu Mifta STLC merupakan singkatan *Software Testing Life Cycle*. Pelatihan alur testing pada perangkat lunak penting agar developer dapat mengetahui proses testing yang harus dilewati.

Pelatihan berikutnya adalah *Testing Methodology*, yang dibawakan oleh bapak Tulus Suyato. Pelatihan ini melanjutkan pembahasan sebelumnya, yaitu mengenai *software testing*. Pada pelatihan kali ini juga dijelaskan macam-macam *testing*. Pelatihan ketiga pada hari ini adalah pelatihan SDLC, Scrum, dan Waterfall yang dibawakan oleh bapak Martin. SDLC merupakan singkatan *Software Development Life Cycle*. Pada pelatihan kali ini dijelaskan metodologi SDLC yang sering digunakan ACC, yaitu Scrum dan Waterfall. Dijelaskan masing-masing dari kedua metode tersebut. Yang terakhir pada hari ini adalah persiapan untuk magang, yaitu *install software* yang akan digunakan dan *setting VPN* agar dapat tersambung dengan *server* milik ACC. Persiapan ini dipandu oleh ibu Astri.

Jumat, 4 Januari 2019

Hari ini kami mendapat pelatihan proses bisnis pada ACC. Pelatihan pertama adalah AA, yaitu *Account Acquisition*, dan EIS (*Enterprise Information Sistem*). Pelatihan ini dibawakan oleh bapak Martin Asril. Pelatihan ini membahas AA sendiri merupakan proses bisnis paling awal, dimana AA adalah proses bisnis yang meliputi dari bagaimana ACC mencari hingga mendapatkan dan menyetujui *customer* yang ada. Pendataan data pribadi *customer* hingga mencapai kontrak.

Pelatihan kedua adalah pelatihan bisnis proses pada AM yaitu AM – COLL (*Account Maintenance - Collection*). Pelatihan ini dibawakan oleh bapak Kevin Christian. Pelatihan kali ini membahas tentang proses pembayaran angsuran *customer*. Pembayaran melalui ACC ini dapat dilakukan melalui *Direct Payment* atau melakukan pembayaran secara langsung kepada ACC, *Maintenance Contract* atau melakukan perubahan kontrak dalam kondisi tertentu, ataupun dengan cara lainnya. Para *customer* juga dapat melakukan pembayaran angsuran tanpa harus datang langsung ke ACC. Pembayaran dapat dilakukan dengan *Transfer*, H2H (*Host to Host*) melakukan pembayaran melalui bank, dan ACC Yes! yang merupakan sebuah aplikasi mobile untuk pembayaran angsuran.

Selain proses bisnis AM, terdapat COLL (*Collection*) yang mengurus bagian penarikan. COLL sebenarnya adalah proses bisnis yang akan dilakukan hanya jika *customer* terlambat melakukan pembayaran angsurannya. COLL sendiri dibedakan menjadi 4, yaitu DC, ARHO, ARRO, dan REMO. DC sendiri dilakukan melalui telepon saat *customer* terlambat membayar selama 1-7 hari. ARHO dilakukan dengan cara mendatangi rumah dari *customer* saat terlambat membayar selama 8-30 hari. Pada proses ARRO barang *customer* sesuai dengan kontrak akan ditarik jika terlambat membayar selama 31-60 hari. Sedangkan REMO akan dilakukannya penarikan paksa menggunakan pihak ketiga.

Pelatihan ketiga adalah pelatihan BO yang dibawakan oleh bapak Rudi. Pelatihan ini membahas tentang BO (*Back Office*). Back Office adalah bagian yang

mengurus tentang pencarian dana ataupun hal lainnya yang juga berhubungan dengan pihak ketiga. Adapun terdapat 4 jenis dari BO yaitu *Join Finance*, *Orafin*, *Collateral* dan *Subrogasi*. Bagian keempat adalah training EIS (*Enterprise Information Sistem*), yang dibawakan oleh Eka Surya. Pada training EIS, diberitahukan mengenai pelaporan dan pencatatan mengenai kegiatan bisnis yang ACC lakukan. EIS sendiri memiliki urutan proses yang dibedakan menjadi 2 bagian yaitu OLTP (*Online Transaction Process*) dan OLAP (*Online Analytical Process*). Pada OLTP sendiri terdapat ETL (*Extract Transform Load*) yang bertugas untuk menyatukan semua data yang akan digunakan ke tahap *Staging*, yang nanti data yang sudah matang akan disimpan ke dalam *Data Warehouse*. Lalu bagian kedua yaitu OLAP, terdapat BI yang bertugas untuk menyusun laporan atas data matang yang tersimpan di *Data Warehouse*. Melalui EIS, terbentuk proses CRM (*Customer Relationship Management*) yang bertujuan untuk terus menjaga dan membangun hubungan dengan para *customer* maupun calon *customer*. Di CRM terdapat 4 cara yaitu, *Tele Service (Inbound Service)*, *Tele Marketing (Outbound Service)*, *CRS (Customer Response Sistem)*, dan *CCRP (Customer Complaint Resolution Progress)*.

Pelatihan terakhir pada hari ini adalah training mengenai *security* di ACC. Pelatihan yang dibawakan oleh bapak Robert menjelaskan pentingnya *security* pada sebuah sistem dan implementasi *security* pada ACC. *Security* ACC bekerja untuk menjaga semua sistem yang ada didalam ACC, didalamnya terdapat 3 buah prinsip utama yaitu *Availability* (ketersediaan), *Confidentiality* (kerahasiaan), *Integrity* (Akurat). Dalam menjaga sistem tersebut, ACC menggunakan beberapa *security device* seperti *Firewall*, *Antispam*, *Antivirus*, *MDM (Mobile Device Management)*, *Identity Management*, dan *WAF (Web Application Firewall)*.

Senin, 07 Januari 2019

Hari ini kami menjalani pelatihan *development program*. Pelatihan hari ini membahas mengenai Outsystem. Outsystem adalah *platform* yang bisa membuat

aplikasi dengan kode yang sedikit, salah satu contoh antar muka *software* ini dapat dilihat di Gambar 2.1. Pada hari ini kami diajari untuk membuat aplikasi pada OutSystem. Pelatihan yang diberikan berupa pelatihan penggunaan aplikasi OutSystem secara mendasar hingga membuat program sederhana. Pada akhir hari kami diuji seberapa kami dapat menangkap pelatihan yang sudah diberikan. Kami diberi tugas untuk membuat aplikasi sederhana untuk transaksi. Saya hanya dapat menyelesaikan 80% dari tugas yang diberikan karena keterbatasan waktu. Tugas atau kuis ini yang akan diberikan hingga akhir pelatihan akan dimasukkan sebagai faktor penilaian.

Selasa, 08 Januari 2019

Hari ini kami diberikan pelatihan mengenai uniPaaS, yang merupakan sebuah aplikasi yang dibuat oleh Magic Software Enterprise. uniPaaS dapat membuat program dengan sangat sederhana (*drag & drop*) namun program tersebut dapat sangat *powerful* seperti yang dapat dilihat pada Gambar 2.2. uniPaaS juga tidak memerlukan banyak penulisan kode. uniPaaS sedikit mirip dengan Visual Studio versi lama, namun uniPaaS lebih mempermudah penggunaanya (terutama pada bagian data). Menurut kata ibu Astri (*developer* ACC di Jakarta) ACC masih menggunakan uniPaaS karena *core system* (transaksi, dll) yang ada di pusat dan cabang sudah diimplementasikan dan sudah dapat berjalan dengan baik, sehingga akan membutuhkan waktu yang sangat lama untuk mengganti penggunaan uniPaaS. Pada hari ini kami diberi pelatihan mengenai uniPaaS secara mendalam, mulai dari cara membuat *blank screen*, tulisan-tulisan, membuat *form*, hingga cara memasukkan data.

Rabu, 09 Januari 2019

Hari ini kami melanjutkan pelatihan uniPaaS. Lanjutan pelatihan yang diberikan adalah membuat tampilan lebih menarik dan penyimpanan data menggunakan PL SQL. Pada akhir hari ini, kami mendapat kuis untuk menguji

pengetahuan dan kemampuan kami dalam menangkap pelatihan yang telah diberikan. Kuis yang diberikan berupa contoh kasus nyata, yaitu membuat sebuah program CRUD untuk cabang ACC, yang dapat melakukan *input* beberapa parameter. *Input* ini akan dimasukan dalam database. Pada program juga diharuskan memiliki *exception* dan beberapa syarat lain. Pada hari ini saya dapat menyelesaikan kuis yang diberikan dengan baik.

Kamis, 10 Januari 2019

Hari ini kami diberikan pelatihan SOFIA. SOFIA merupakan bahasa singkatan yang digunakan developer ACC. SOFIA yang sebenarnya adalah *software* AG Designer. Salah satu kegunaan SOFIA adalah membuat *database* berbasis SQL, dengan menggunakan *flowchart* seperti yang terlihat pada Gambar 2.3. Jadi melalui SOFIA dapat membuat PL SQL secara GUI. Pelatihan yang kami dapatkan berupa pembuatan *flowchart database* untuk pengambilan dan pembuatan data. Pada akhir hari kami mendapat kuis. Pada kuis kali ini, kami diminta untuk membuat sebuah *flowchart* yang akan dapat menangani proses bisnis penjualan mobil. Pada kuis kali ini saya dapat menyelesaikan meskipun kurang pada beberapa bagian (kondisi dan sebagainya).

Jumat, 11 Januari 2019

Hari ini kami mendapat training mengenai Laravel. Laravel merupakan *framework* PHP yang digunakan untuk mempermudah *developer* dalam membuat sebuah *website* berbasis PHP, dengan *welcome page* seperti pada Gambar 2.4. Berdasarkan *survey*, Laravel adalah *framework* yang paling banyak digunakan dalam pembuatan *website*. Pada pelatihan kali ini, kami diajarkan untuk membuat *website* sederhana yang mampu menjalankan fungsi CRUD: *create*, *read*, *update*, *delete*. Untuk database yang digunakan, kami menggunakan MySQL. Lalu, dalam mengisi SQL dengan data *dummy* kami diajarkan menggunakan *faker*. Pada akhir hari, kami diberi kuis untuk melatih pemahaman kami. Kuis ini kami disuruh untuk

membuat *website* yang mampu menangani CRUD, *login*, terdapat halaman profil dan dapat meng-*upload* gambar.

Senin, 14 Januari 2019

Hari ini kami menjalani pelatihan mandiri. Seperti yang sudah dijelaskan sebelumnya, OutSystem memiliki *channel* yang berisikan video pelatihan untuk *Development Web* dan *Development Mobile Application*. Dalam satu rangkaian pelatihan, biasanya terdapat 20 lebih video dengan durasi pelatihan mulai dari 20 jam. Pada hari ini saya mengerjakan sejumlah 13 pelatihan yang berisikan *tutorial* dasar untuk membuat aplikasi *website* sederhana menggunakan Outsystem. Pelatihan saya lanjutkan dirumah meskipun saya tidak menyelesaikan seluruh video pelatihan tersebut. Yang saya dapatkan dari melihat video pelatihan tersebut adalah cara *setup environment* termasuk persiapan komponen-komponen dan *packages*, menggunakan *template*, membuat *pop up*, *link navigation*, dan *debug*.

Selasa 15 Januari 2019

Hari ini, saya melanjutkan pelatihan mandiri OutSystem. Jika pada hari senin *platform* yang saya kerjakan adalah *website*, maka hari ini saya melakukan pelatihan pada *platform Mobile*. Yang saya coba kerjakan pada hari ini adalah 15 video *tutorial mobile application* awal OutSystem. Dalam video tersebut saya mempelajari beberapa hal yang cukup mirip dengan *platform website*. Beberapa hal yang berbeda adalah cara *deploy*, *navigation bar*, keamanan PIN, tampilan-tampilan, dan hal yang lain. Hingga akhir hari ini saya masih belum bisa menyelesaikan seluruh *tutorial* nya.

Rabu, 16 Januari 2019

Hari ini berbeda dengan pelatihan sebelumnya. Pada hari ini kami diberikan training PL SQL. PL SQL merupakan *Procedural Language for SQL*. Secara singkat PL SQL mampu memberikan data yang diminta hanya dengan memanggil nama prosedur SQL dan parameter-parameter yang dibutuhkan. Pada training ini kami dilatih untuk dapat membuat sebuah PL SQL pada komputer ACC pusat. Kami menggunakan VPN Forti Client dengan antar muka seperti yang terlihat di Gambar 2.5 dan software *remote* bawaan Windows untuk dapat menggunakan komputer ACC. Terlebih dahulu dibuat tabel yang merupakan tabel yang berisikan tentang merk mobil, nama mobil, tanggal *create/update*, dan pengguna yang melakukan *create/update*. PL SQL yang dibuat adalah PL SQL untuk melakukan *select*, *create*, dan *update* pada tabel yang dibuat sebelumnya.

Kamis, 17 Januari 2019

Hari ini pelatihan PL/SQL dilanjutkan. Ibu Awa meminta untuk menyelesaikan PL/SQL yang kemarin belum selesai dan *submit* bagi yang sudah selesai. Karena saya sudah selesai, saya *submit* pekerjaan saya. Setelah saya *submit* saya memutuskan untuk kembali belajar PL/SQL. Pada sore hari kami diberikan pelatihan lanjutan. Kami diminta untuk membuat tabel dengan *flow* seperti pada Gambar 2.6, yang berisikan tentang cabang, kode cabang, status cabang, dan beberapa hal *credential* lain. Berikutnya kami diminta untuk melakukan *insert* data-data mengenai cabang. Setelah itu kami diminta untuk membuat fungsi untuk menampilkan nama cabang dan beberapa status lain berdasarkan *input* kode cabang. Prosedur yang diminta adalah prosedur untuk mengecek *input* kode cabang lalu mengecek status-status terkait dan memberikan status pada cabang tersebut. Saya tidak berhasil menyelesaikan kuis, hanya tinggal pemberian kondisi/status pada cabang.

Jumat, 18 Januari 2019

Hari ini pelatihan SQL dan pelatihan OutSystem dilanjutkan secara mandiri karena *supervisor* kami Ibu Awa sedang berhalangan hadir. Akhirnya saya mampu menyelesaikan kuis tersebut. Setelah *submit* kuis, saya melanjutkan pelatihan OutSystem. Saya hanya melihat video tutorial tentang *environment* OutSystem dan fungsionalitas program (*shortcut*, kemampuan program).

Senin, 21 Januari 2019 - Jumat, 25 Januari 2019

Hari ini saat pembagian *project*, saya dan Michael Budi mendapatkan *project* Ionic-Laravel. Pada proyek kali ini, kami diminta untuk membuat sebuah aplikasi bernama AccWhistle yang sebelumnya sudah dibuat di *Website* dengan *framework* Laravel yang dapat dilihat pada Gambar 2.7, kemudian di ubah ke *Web View* untuk android, namun pihak *Vendor* kurang puas dengan hasil *Mobile* tersebut, sehingga mereka meminta dibuatkan ulang dari awal. Dikarenakan saya belum pernah menggunakan *framework* Ionic, dan pada masa pelatihan magang kemarin juga tidak ada pelatihan mengenai Ionic, maka BSA *project* ini, bapak Ardy, memberikan kami waktu selama seminggu untuk mempelajari *framework* ini dan mencoba coba *tutorial* yang ada di forum Ionic serta Youtube, hal ini juga dikarenakan bapak Ardy sedang sakit dan akan menjalan operasi, sehingga sampai saat bapak Ardy sehat kembali kami diminta mempelajari *framework* yang akan digunakan pada project ini, yaitu Ionic dengan logo seperti pada Gambar 2.8 dan Laravel.

Senin, 28 Januari 2019

Setelah seminggu sebelumnya saya belajar dan berlatih mengenai *framework* Ionic, saya rasa saya sudah cukup memahami *framework* ini secara fungsionalitas dan struktur. Maka hari ini saya mencoba mendalami, kerangka kerja *front-end* pada *framework* ini, yaitu AngularJS. Jadi *framework* Ionic yang akan

saya kerjakan ini, memiliki *back-end* yang menggunakan bahasa TS (TypeScript) dimana bahasa ini berbasis JavaScript, yang dikembangkan oleh Microsoft, dengan menambahkan fitur *strong-typing* & konsep pemrograman OOP (*Object Oriented Programming*) klasik. Dan *front-end* yang digunakan oleh *framework* ini adalah HTML dengan AngularJS dan CSS. Pada HTML di Ionic, kita dapat menggunakan kode dan *tag* dari Ionic itu sendiri dan juga dari AngularJS. Untuk *tag* dari Ionic di HTML ini, sebagian besar mengarah ke tampilan yang akan muncul nantinya, sebaliknya lagi, *tag* dari AngularJS disini lebih ke arah *back-end* seperti halnya JavaScript.

Selasa, 29 Januari 2019

Hari ini saya membagi tugas dengan Michael Budi, dikarenakan ada 2 *platform*, maka kami memutuskan untuk membagi *framework* yang dikerjakan, saya menyerahkan pada Budi untuk memutuskan *framework* apa yang akan dia pilih, dan *framework* yang tidak dia pilih adalah *framework* yang akan saya kerjakan. Disini Budi memilih untuk bekerja pada *framework* Ionic, sehingga saya bekerja pada *framework* Laravel. Ionic disini nantinya akan digunakan untuk membuah sebuah aplikasi utuh berbasis Android dan iOS, sedangkan Laravel akan meng-*clone* dari *project web*, dan saya hanya akan membuat API agar semua data yang sudah ada di *web* AccWhistle dapat dipakai dan disinkronisasikan dengan aplikasi *mobile* AccWhistle. Karena saya tidak begitu familiar dengan Laravel sebelumnya, dan ini baru pertama kali saya menggunakan Laravel, maka hari ini saya mencoba mencari *tutorial* terlebih dahulu mengenai apa itu Laravel, dan apa yang perlu diketahui sebelum membuat *project* pada *framework* ini.

Rabu, 30 Januari 2019

Hari ini dokumen TD (*Technical Design*) dan FD (*Functional Design*) dari *project* AccWhistle *mobile* ini berikan kepada kami. Dan kamipun diminta oleh bapak Ardy untuk membaca dan memahami dokumen tersebut. Dokumen FD

berisikan *flowchart* bagaimana aplikasi ini akan berjalan, dan dokumen FD berisikan fungsi fungsi apa yang dibutuhkan oleh aplikasi ini. Dan karena beberapa hal seperti bagaimana alur pelaporan di ACC berlangsung, karena *project* AccWhistle ini adalah aplikasi untuk pengguna (*customer* ACC) dapat melaporan atau menyampaikan keluhan terhadap divisi atau karyawan yang ada di ACC, maka saya meminta bapak Ardy menjelaskan alur pelaporan yang terjadi dan arti dari tiap *flag* yang diberikan pada setiap data yang disimpan pada *database*.

Kamis, 31 Januari 2019

Hari ini untuk pertama kali nya saya mengimplementasikan apa yang telah saya pelajari pada *tutorial* sebelumnya ke dalam *project* AccWhistle ini. Berdasarkan permintaan dari teman saya Budi, saya membuat API untuk *login* seperti pada Gambar 2.9, karena halaman pertama yang akan dibuat adalah *login*. Dengan menambahkan fungsi baru pada AuthController yang bernama *login*. Di fungsi ini saya menggunakan Facade atau kelas bawaan Laravel yang bernama *Auth*, dimana di kelas ini sudah disediakan kode untuk autentifikasi dan kita hanya tinggal menggunakannya saja, pada halaman *login*, terdapat 2 variable yang digunakan untuk *login*, yaitu *email* dan *password*, kedua parameter ini lah yang nantinya akan di lempar, saat *link* API di *hit*. Lalu dengan menggunakan Routes pada Laravel, saya mengarahkan *flow* aplikasi ini ke arah AuthController, ketika *link* API untuk *login* di *hit*. Lalu dengan fitur *Auth* bawaan Laravel, saya mengecek apakah pengguna dengan *email* dan *password* tersebut ada di *database* atau tidak, jika ada maka balikan atau *response* ketika *link* API di *hit* adalah JSON yang berisikan data dari pengguna tersebut, namun bila data pengguna tidak ditemukan, maka balikannya adalah *error* dengan pesan “*Your credential is wrong*”.

Jumat, 1 Februari 2019

Setelah sebelumnya saya membuat API untuk *login*, hari ini Budi meminta saya untuk membuat API *register*. Sama seperti sebelumnya, saya membuat kelas

baru di AuthController, dan kemudian membuat sebuah fungsi baru bernama *register*, dengan parameter yang dibutuhkan ketika API di *hit* adalah, *fullname*, *no_handphone*, *email*, dan *password*. Di dalam fungsi yang saya buat ini, terdapat validasi nilai yang akan dibuat menjadi baris atau *row* baru pada tabel, dengan menggunakan fungsi bawaan Laravel yang bernama *validate*. Dalam validasi ini ketentuan agar data dapat diterima dan dibuat menjadi sebuah baris baru adalah *fullname* harus ada, berformat *string*, dengan minimal 6 karakter dan maksimal 255 karakter dengan *regex* `/^[pL\s]+$/u` yang mana berarti hanya *Latter* atau huruf saja yang dapat diterima, karakter spesial dan angka akan ditolak, kemudian *no_handphone* yang bersifat wajib, dan *regex* `/(08)[0-9]{9}/` yang berarti hanya angka saja yang dapat diterima, selanjutnya *email* yang juga wajib dan harus berformat email, unik, yang berarti hanya ada 1 email dan tidak boleh ada lagi yang sama, dan yang terakhir adalah *password* dimana wajib dan harus setidaknya ada 6 karakter. Dan untuk parameter atau *attribute* pada tabel yang lain, saya buat *default* dengan memasukkan nilainya langsung secara *back-end* atau dari fungsi yang saya buat, diantaranya adalah parameter *group* (*default* adalah *member*) dan *flag_active* (*default* nya adalah 0). Ketika data nya masuk ke dalam *database* maka email verifikasi akan dikirimkan ke email yang telah didaftarkan. Apabila pengguna melakukan verifikasi email, maka attribute *flag_active* yang sebelumnya 0 berubah menjadi 1, ketika itulah pengguna ini dapat dipakai untuk *login*. Karena dalam *login* terdapat pengecekan terakhir setelah pengguna ditemukan, yaitu apakah *flag* nya 0 atau 1. API *register* sendiri dapat dilihat pada Gambar 2.10.

Senin, 4 Februari 2019 - Rabu, 6 Februari 2019

Dua hari ini saya berfokus untuk mempelajari lagi lebih dalam bagaimana API dapat bekerja di Laravel, karena 2 API yang saya buat sebelumnya tergolong mudah, dan ada banyak contoh yang bisa langsung dapat digunakan di internet, karena saya sadar selanjutnya API yang akan saya buat akan lebih kompleks lagi. Apa yang saya pelajari di 2 hari ini lebih menambah pehaman saya kepada Routes yang ada, yang mana apa yang saya pahami adalah Routes disini berfungsi penting

untuk menentukan alur kemana data yang diterima oleh API akan diarahkan ke fungsi *back-end* yang sesuai di Controllers Laravel.

Kamis, 7 Februari 2019 - Jumat, 8 Februari 2019

Berdasarkan permintaan bapak Ardhy, saya diminta menambahkan sebuah token pada *database project* ini, yang mana token ini akan berfungsi penting untuk mencegah 2 *device login* pada 1 akun dan 1 waktu yang sama. Dan hari ini juga dimana kak Wira, yaitu mahasiswa UAJY angkatan 2014, dimasukkan kedalam team kami untuk membantu karena *project* dia sebelumnya yaitu Acc Hunter telah selesai, dan juga karena kak Wira lah yang membuat *project* AccWhistle untuk *website*. Karena saya mengalami kebuntuan dengan bagaimana mengimplementasi token ini pada *database* dan juga Laravel, maka kak Wira meng-*handle task* ini, sehingga pada hari Jumat, sistem token ini sudah bisa diimplementasikan dan digunakan untuk *project* AccWhistle.

Senin, 11 Februari 2019 - Senin, 18 Februari 2019

Berdasarkan apa yang telah saya pelajari tentang API di hari-hari sebelumnya, maka hari ini sampai 1 minggu kedepan saya berfokus untuk membuat semua API yang dibutuhkan oleh aplikasi yang dibuat pada *framework* Ionic, baik itu yang diminta oleh Budi, atau yang belum dia minta sekalipun, saya akan menyediakannya. API pertama yang saya siapkan adalah untuk tabel *Categories*, tabel ini berisikan kategori pelaporan yang akan dilakukan oleh pengguna dalam konteks yang umum. Di dalam *CategoriesController* ini saya membuat 3 buah fungsi yaitu *addCategory*, *getAllCategory* dan *deleteCategory*, dan yang akhir nya dipakai hanyalah fungsi *getAllCategory* (Gambar 2.11), karena aplikasi Android yang kami buat hanya akan membaca kategori apa saja yang ada untuk pelaporan.

Selanjutnya adalah *FaqController*, di *Controller* ini terdapat 3 fungsi, yaitu *createFAQ*, *getAllFAQ*, dan *deleteFAQ*. Sekali lagi, fungsi yang digunakan pada

API hanya *getAllFAQ*, dimana fungsi ini akan dipanggil pada halaman FAQ di aplikasi AccWhistle, dan ketika API ini di hit seperti pada Gambar 2.12, maka *response* nya adalah JSON dengan data yang berisikan pertanyaan dan jawaban yang sudah dimasukkan ke tabel FAQ di *database*.

API ketiga yang saya buat adalah untuk tabel SubCategories, dimana semua fungsi nya di simpan pada SubCategoriesController dan berisikan 3 fungsi, yaitu *addSubCategory*, *getAllSubCategory* dan *deleteSubCategory* dan fungsi yang digunakan sebagai API adalah *getAllSubCategory*. Ketika API ini di *hit* seperti pada Gambar 2.13, maka *reponse* yang diberikan adalah semua nama dan deskripsi SubCategory serta *foreign key* yang diperoleh dari *primary key* tabel *Categories*.

Terakhir ada API untuk laporan yang pengguna buat atau kami sebut sebagai *ticket* atau tiket. Dengan nama TicketController, controller ini tergolong *core controller* yang ada di *project* ini, karena berisikan sebagian besar fungsi untuk meng-*handle* pelaporan yang mana pelaporan adalah hal utama yang dilakukan oleh aplikasi AccWhistle ini. Fungsi-fungsi yang ada didalam *controller* ini diantaranya *getMyTicketList* (Gambar 2.14) untuk mendapatkan semua data laporan yang telah dilakukan sebelumnya, *readTicket* untuk mengubah *flag read* status dari tiket yang dibuat dari N (*No*) menjadi Y (*Yes*), *getTicket* untuk mendapatkan detail dari pelaporan yang telah dilakukan, *upload* untuk meng-*upload* foto yang akan mendukung laporan yang dilakukan, *getTicketIdentifier* untuk mendapatkan kode *id* unik terbaru dari kolom *id_ticket* kemudian ditambahkan 1 (format dari *id_ticket* adalah PLPyyyymmddxxx), *create* untuk membuat laporan baru (didalam fungsi ini, fungsi lain seperti *getTicketIdentifier* dan *upload* di panggil), dan beberapa fungsi *get* untuk menarik data detail *ticket* dengan *flag* tertentu seperti, hanya data yang ditindak lanjutin, diterima, dianalisa, *pending* dan *closed*.

Selasa, 19 Februari 2019 - Jumat, 22 Februari 2019

Hari ini saya telah membuatkan banyak API terlepas dari akan digunakan atau tidak nantinya. Dan apabila data yang dibalikkan saat API di *hit* tidak sesuai

atau perlu penyesuaian lebih lanjut maka saya hanya perlu mengubah sedikit fungsi yang telah saya buat sebelumnya. Jadi hari ini sampai 3 hari kedepan saya membantu Budi untuk membuat tampilan *front-end* untuk aplikasi AccWhistle di *framework* Ionic.

Saya membantu membuat antar muka halaman FAQ (Gambar 2.15), dimana dihalaman ini, semua pertanyaan akan dibuat menjadi sebuah list, yang nanti nya ketika pengguna mengklik pertanyaan yang ada, maka jawaban dalam akan muncul dibawah pertanyaan tersebut dengan animasi *drop down*. Kedua adalah halaman profil (Gambar 2.16), dihalaman ini hanya akan berisikan tombol *login* dan *logout* untuk pengguna. Ketiga adalah halaman daftar tiket dari pengguna, untuk halaman ini (Gambar 2.17), dibuat oleh kak Wira, karena untuk semua yang berhubungan dengan daftar tiket dan detail tiket menjadi *task* atau tugas kak Wira. Keempat adalah halaman *forget password*, *forget password* ini terbagi menjadi 2 halaman, pertama adalah untuk memasukkan email yang telah di daftarkan, ketika email telah di input dan pengguna menekan tombol *submit* (Gambar 2.18), maka email akan dikirimkan ke pada pengguna berdasarkan email yang tadi pengguna masukkan, lalu jika pengguna mengklik tombol reset yang ada dalam email tersebut (Gambar 2.19), maka pengguna akan di arahkan ke halaman baru dimana pengguna memasukkan password baru dan konfirmasi password baru, setelah form tersebut di submit, maka password pengguna dalam database akan di ganti dengan yang baru.

Senin, 25 Februari 2019

Hari ini saya dengan Budi fokus mengerjakan *home page* (Gambar 2.20) dari aplikasi AccWhistle ini. Karena tampilan halaman ini lebih kompleks dari pada halaman yang lain. Bapak Ardy sudah memberikan kami *mockup* dari *home screen* yang akan dibuat berdasarkan persetujuan dengan *vendor*. Tampilan yang dibuat sedikit mirip dengan tampilan yang ada di *website* secara pemilihan warna dan gambar yang ada karena kami berusaha tetap mengikuti konsep dasar tampilan dari

AccWhistle ini baik di *website* maupun aplikasi *mobile*. Saat sore hari setelah kami selesai membuat tampilan dari halaman ini, lalu kami *push project* ini ke gitlab agar dapat di *pull* untuk di *review* oleh bapak Ardy keesokan harinya.

Selasa, 26 Februari 2019

Setelah sebelumnya saya seharian dengan Budi membuat tampilan dari *Home Page*, hari ini pagi-pagi sekali BSA kami bapak Ardy menghubungi kami dan mengatakan bahwa ada perubahan terkait tampilan dari halaman *Home* ini, karena setelah memberikan hasil tampilan yang kami buat kemarin kepada *vendor*, beliau merasa sedikit kurang puas dengan tampilan yang ada. Bapak Ardy pun membuat *mockup* tampilan yang baru, berdasarkan *request* dari *vendor* juga tentunya. Dan kami seharian ini berusaha mengubah kembali tampilan yang telah kami buat agar sesuai dengan *mockup* baru yang diberikan BSA kepada kami.

Rabu, 27 Februari 2019 - Jumat, 1 Maret 2019

Setelah kami membuat semua halaman yang dibutuhkan untuk saat ini, maka yang saya dan Budi lakukan untuk beberapa hari kedepan adalah menyesuaikan *style CSS* dari setiap halaman yang telah kami buat HTML nya agar tampilan yang ada menjadi lebih baik dan *responsive* terhadap ukuran layar dari *device* pengguna. Karena saya dan Budi belum begitu paham mengenai CSS kami pun melakukan research terlebih dahulu di hari pertama *task* tentang *style CSS* ini. Bagaimana cara menggunakan nya, bagaimana menggunakan sebuah *style* hanya di satu bagian dari sebuah halaman, dan beberapa hal lain nya. Dan di *task* ini juga kami mendapatkan ilmu baru berharga lainnya.

Salah satu yang paling saya ingat adalah untuk *framework* Ionic, *tag* HTML yang kami gunakan itu tidak seperti *tag* HTML pada umumnya, setiap *tag* yang kami gunakan diawali dengan *<ion-*, contohnya seperti *<ion-content>* yang mana dalam HTML biasa itu adalah *<body>*, lalu ada *<ion-card>* untuk membuat

tampilan menjadi seperti sebuah *card*, ada *<ion-item>* seperti hal nya tag *<p>* pada HTML biasa. Semua perbedaan itu membuat kami kesulitan dengan menerapkan CSS yang telah kami buat, karena terkadang *style* yang kami buat tidak memberikan efek apapun pada HTML yang ada. Sehingga bapak Ardy mengajarkan bahwa dalam Ionic, cara terbaik untuk mendapatkan kelas dari sebuah tag HTML adalah dengan melakukan *inspect element* saat aplikasi tersebut di jalankan di *browser* menggunakan perintah *ionic serve* pada *terminal*. Contoh nya adalah ada sebuah tag *<ion-card class="card-welcome">* (Gambar 2.21), apabila biasanya kelas yang dibuat di CSS adalah *.card-welcome* maka apabila kita melakukan *inspect element* bagian itu di *browser*, maka kelas *style* yang muncul di *browser* adalah *ion-card.card-welcome.card* dan kelas itu lah yang kami pakai untuk menjadi kelas pada CSS yang kami buat, sehingga sekarang apabila kami melakukan *re-run* aplikasi yang kami buat ini pada *browser*, maka *style* yang telah kami buat sekarang memberikan efek *style* seperti sebagaimana mestinya.

Senin, 4 Maret 2019 - Selasa, 5 Maret 2019

Selama 2 hari saya diberikan *task* oleh bapak Ardy untuk membuat *back-end* dari halaman lapor (Gambar 2.22) yang mana merupakan halaman inti dari aplikasi ini. Berdasarkan *mockup* dari halaman lapor, saya membuat parameter-parameter yang dibutuhkan untuk di kirimkan melalui API yang sebelumnya telah saya buat. Di halaman ini saya melakukan 4 pemanggilan API, 2 API pertama yang di *hit* ketika halaman akan dibuka adalah *getAllSubcategory* dan *getAllCategory*. Selanjutnya adalah API untuk melakukan *upload*, pada halaman lapor ini, pengguna dapat memilih gambar (jpg,gif,png,jpeg), dokumen (doc,pdf,docx,xls,xlsx,ppt,pptx) atau media (mpga,mp3,mpeg,mp4,3gp) yang sekiranya dapat mendukung laporan yang akan diberikan. Sebelum API *upload* ini di *hit*, saya membuat kode untuk membatasi tipe *file* yang akan di *upload* (Gambar 2.22) seperti yang telah saya paparkan sebelumnya. Terakhir terdapat sebuah kondisi khusus, dimana jika pengguna sudah *login*, maka yang tadi nya pengguna perlu menginputkan email, sekarang kolom email sudah secara otomatis terisi

dengan email dari pengguna yang *login* dan dibuat menjadi *disable* agar pengguna tidak dapat mengedit teks email tersebut (Gambar 2.23).

Rabu, 6 Maret 2019

Hari ini saya membuat *front-end* atau tampilan dari lapor *page* berdasarkan *mockup* yang telah diberikan bapak Ardy dan saya menghubungkan semua parameter yang di HTML agar dapat memanggil semua fungsi yang telah saya buat sebelumnya di file *.ts* (*type script*) untuk halaman lapor ini, hasil nya dapat dilihat pada Gambar 2.24.

Jumat, 8 Maret 2019

Hari ini setelah apa yang telah saya dan Budi kerjakan sebelum nya, mas Ardy menemukan beberapa *minor bug* yang ada di aplikasi AccWhistle, seperti *tab* aktif dari setiap halaman, apabila kami berganti ke halaman tertentu, *tab* aktif tersebut tidak ikut berganti dan tetap aktif pada halaman sebelumnya, kemudian juga tombol *login* yang ada di halaman *home* tidak berubah menjadi tombol lapor sekalipun pengguna telah melakukan *login*, juga ketika pengguna melakukan *login*, halaman *login* tidak *ter-dismiss* atau tampilan utama tidak berpindah ke halaman *home* namun tetap di halaman *login*.

Senin, 11 Maret 2019

Setelah sebelumnya pada halaman lapor, kami menggunakan data *dummy* yang dimasukkan *manual* pada *back-end* aplikasi, maka hari ini kami melakukan implementasi *hit* API pada halaman lapor untuk API *getAllCategory* dan *getAllSubCategory*. Semua API yang akan dipanggil akan di *hit* didalam sebuah fungsi bawaan Ionic yaitu *ionViewDidLoad()*. Kelas ini merupakan kelas khusus bawaan Ionic dimana semua kode yang ada di dalam kelas ini akan di jalankan

bahkan sebelum tampilan dari halaman tertentu berhasil diproses. Karena ketika halaman selesai diproses maka menu *drop down* dari kategori sudah harus berisikan daftar kategori yang dapat dipilih oleh pengguna.

Selasa, 12 Maret 2019

Hari ini saya ditugaskan bapak Ardy untuk mengimplementasikan salah satu fitur modern yang ada saat ini, yaitu DeepLink kedalam *project* AccWhistle. Fitur DeepLink ini bertugas untuk membuka aplikasi melalui *browser*. Ide nya adalah ketika pengguna melakukan pelaporan, *reset password*, dan *register*. Pengguna akan mendapatkan email yang berisikan sebuah tombol, apabila biasanya tombol tersebut apabila di klik maka akan memanggil API tertentu untuk mengubah data langsung ke *database*, seperti jika *pengguna register*, maka pengguna akan mendapatkan email verifikasi, dan ketika tombol verifikasi pada email tersebut di tekan, maka API untuk verifikasi pengguna akan di panggil dan browser akan terbuka yang kemudian tidak memunculkan apa-apa karena pengguna melakukan verifikasi dari aplikasi AccWhistle dan bukan *website*. Dengan menerapkan DeepLink dalam suatu *project* aplikasi *mobile* baik itu Android maupun iOS, maka setiap aplikasi akan memiliki *address* sendiri, dimana apabila pengguna tau *address* tersebut, pengguna dapat membuka aplikasi dari *browser* apapun. Namun karena DeepLink merupakan fitur *back-end* jadi *address* ini hanya diketahui oleh *developer* dari setiap aplikasi. Dengan itu pula saya mengubah yang tadi nya saya hanya memanggil API, saya tambahkan beberapa baris kode dimana ketika API telah selesai dipanggil dan perubahan sudah di lakukan di *Controller*, maka Routes akan *men-direct browser* untuk membuka *address* dari aplikasi tersebut, dan aplikasi AccWhistle pun akan terbuka dan *browser* akan tertutup dengan sendirinya.

Pada *project* Ionic saya juga menambahkan beberapa kode dimana *address* yang di gunakan untuk membuka aplikasi dapat juga membawa beberapa parameter setelah / (*slash*). Contohnya saja saya memanggil AccWhistle/resetEmail, yang mana terdapat parameter bernama *resetEmail* yang nantinya akan berisikan email

dari pengguna yang akan di *reset password* nya, dan parameter ini dapat ditangkap oleh aplikasi ketika aplikasi mulai berjalan dan dioperkan ke halaman tertentu dimana parameter ini diperlukan, dalam contoh ini yaitu halaman *reset password*, karena dihalaman ini pengguna tidak menginputkan email, hanya menginputkan *password* dan konfirmasi *password*. Jadi DeepLink merupakan sebuah fitur dimana *developer* dapat mengontrol aplikasi dari pihak ke dua dimana dalam kasus ini adalah membuka dan mengoper nilai dalam parameter dari sebuah email, bahkan ketika aplikasi buatan *developer* tersebut tidak dibuka sebelumnya.

Rabu, 13 Maret 2019

Hari ini saya mengoptimisasi tampilan dari halaman lapor, dimana teks dihalaman ini terkadang tertimpa dengan kolom *input* apabila dibuka dengan ukuran layar *device* yang lebih besar. Dan kami juga menambahkan deskripsi dari kolom yang akan di isi agar pengguna tau apa yang harus dituliskan dalam laporannya.

Kamis, 14 Maret 2019

Setelah sebelumnya saya membuat *limit* ekstensi *file* yang dapat di pilih oleh pengguna dalam pelaporannya. Maka sekarang saya membuat *limit* dari ukuran *file* yang akan di pilih oleh pengguna sebesar 10 MB saja, karena apabila lebih dari itu maka *server* pusat dikhawatirkan tidak akan sanggup menyimpan banyak nya laporan dengan file tambahan yang berukuran besar. Pembatasan ukuran ini akan berjalan ketika pengguna telah memilih *file* pada *file manager* di *device* milik nya, sistem akan mengambil ukuran *meta file* dan kemudian dibandingkan dalam *if*, apabila lebih besar dari 10 MB maka pengguna akan secara otomatis dipaksa memilih *file* lagi (Gambar 2.25).

Jumat, 15 Maret 2019

Hari ini berdasarkan *review* dari bapak Ardy, terdapat kesalahan penamaan *file* ketika melakukan *upload file*, karena itu kami berusaha memperbaikinya, kesalahan ini terdapat pada fungsi *upload* yang ada pada *TicketController* di Laravel. Sebelum diubah, sistem akan mendapat nama original dari *file* yang dipilih pengguna, oleh karena itu saya ubah dengan menetapkan nama berdasarkan format yang ditentukan yaitu menggunakan *id* dari tiket.

Senin, 18 Maret 2019 - Rabu, 20 Maret 2019

Hari ini sampai 2 hari kedepan saya melakukan *fixing bug* dari sisi *developer* sebelum diberikan pada QC (*Quality Control*) untuk dilakukan pengujian *bug*. *Bug* yang kami telah selesaikan pada 3 hari ini adalah *SubCategory* yang di tampilkan setelah memilih *Category* bukan merupakan bagian *Sub* dari *Category* tersebut, jadi terdapat kesalahan data yang akan ditampilkan. Kami selesaikan dengan memeriksa parameter yang berisikan *foreign key* yang diterima dari *response hit* API, dan ternyata kami salah menggunakan parameter untuk pembanding pada HTML, sehingga yang seharusnya ditampilkan adalah data yang memiliki *foreign key* yang sama dengan *primary key Category*, namun yang ditampilkan adalah data *SubCategory* yang memiliki *primary key* sama dengan *primary key Category*. Dan beberapa *bug* lainnya adalah *bug* pada bagian validasi data dibagian Ionic, terdapat kesalahan jumlah minimal karakter, format *regex* yang dipakai, dan beberapa kesalahan validasi lainnya.

Kamis, 21 Maret 2019

Bapak Ardy hari ini meminta saya untuk membuat sistem *multi upload*, jadi yang sebelumnya hanya 1 *file* saja yang dapat pengguna *upload* dalam 1 laporan, maka sekarang pengguna dalam memilih lebih dari 1 *file* yang akan di *upload* untuk

mendukung laporan pengguna tersebut. Dalam hal ini saya menggabungkan kode kode pada bagian *back-end* dan *front-end*, untuk *back-end* sendiri saya hanya menggunakan *array* yang terus di *increment* setiap sebuah *file* telah berhasil dipilih dan tentunya telah melewati validasi untuk *file*, *array* ini saya gunakan untuk menyimpan setiap *meta* data dari *file* yang pengguna pilih seperti nama, *file name*, dan ekstensi *file*. Lalu kemudian ketika laporan akan di *submit*, maka setiap *array* ini akan di baca dalam sebuah *for loop* yang berdasarkan jumlah isi *array* ini, dan dalam setiap akhir *loop upload* API akan di *hit* dan *file* akan di simpan dalam *server*. Untuk *front-end* disini, saya menggunakan salah satu *loop* bernama *ngFor* dari AngularJS (Gambar 2.26), setiap pengguna selesai memilih sebuah *file*, maka *file selector* lainnya akan muncul. Dan karena terdapat perubahan sistem *upload file*, maka penamaan *file* dalam *server* juga ikut berubah. Tetap menggunakan format penamaan sebelumnya, yaitu menggunakan nama *file* yang sama dengan nama *id* tiket, namun diikuti dengan jumlah *file*, seperti *filename_0* untuk *file* pertama, *filename_1* untuk *file* kedua, dan seterusnya.

Jumat, 22 Maret 2019 - Jumat, 29 Maret 2019

Selama seminggu penuh kami memasuki masa *bug fixing* berdasarkan *bug* yang ditemukan oleh QC, berikut terdapat daftar 10 *bug* yang ditemukan (Gambar 2.27), serta beberapa permintaan personal dari pegawai QC yang memeriksa aplikasi kami dan upaya kami dalam menyelesaikan diantaranya :

1. Menu Error : *Input No Handphone* (Gambar 2.28)

Deskripsi : Pada insert no hp masih bisa menggunakan huruf, harusnya angka saja.

Penyelesaian: Kami menambahkan *regex* yang hanya akan menerima inputan dari keyboard berupa angka saja, *regex* nya adalah *^[a-zA-Z]**.

2. Menu Error : *Administrator Change Email* Pengguna (Gambar 2.29)

Deskripsi : Ketika admin ganti email pengguna, tidak ada konfirmasi verifikasi ulang ke pengguna, dimana berdampak pada pengguna yang tidak mengetahui perubahan tersebut tdk dapat *login*/lapor dan bersangkutan juga kepada laporan tidak akan masuk ke email pengguna.

Penyelesaian: Bapak Ardy mengatakan kepada QC bahwa *Administrator* tidak memiliki alasan untuk mengubah email dari pengguna yang telah terdaftar sehingga permintaan ini dianggap selesai.

3. Menu Error : Menu *Account* (Gambar 2.30)

Deskripsi : *Request avatar* di *account* bisa pake foto, karena di menu *account* sudah simple.

Penyelesaian: Bapak Ardy mengatakan kepada QC bahwa pengguna tidak memiliki *profile picture*, hal ini juga diterapkan di *AccWhistle website*, sehingga apabila aplikasi *mobile* menerapkan *profile picture* maka *website AccWhistle* juga harus menerapkan hal serupa dan itu membutuhkan waktu lagi, sehingga permintaan ini dianggap selesai.

4. Menu Error : Menu *Login* (Gambar 2.31)

Deskripsi : Pada menu *login* ada keterangan "belum punya akun?" dan "daftar disini" lebih baik di gabung karena jika tidak tulisan "belum punya akun" akan rancu karena tidak ada perbedaan yang bisa di klik dan tidak.

Penyelesaian: Kami selesaikan dengan memberikan warna pada setiap teks yang bisa pengguna klik. Sebelum nya teks "belum punya akun ?", "daftar disini", dan "lupa password ?" berwarna hitam, kini teks "daftar disini" menjadi berwarna biru dan "lupa password ?" berwarna merah (Gambar 2.32).

5. Menu Error : *Loading* (Gambar 2.33)

Deskripsi : *Loading logout* suka lama.

Penyelesaian: Setelah kami mencari penyelesaian dari permasalahan ini dan tidak terdapat masalah seperti yang disampaikan oleh QC, maka kami menyampaikan kepada bapak Ardy bahwa kemungkinan bahwa koneksi yang terdapat di kantor ACC mengalami masalah karena koneksi kantor ACC sudah diberi batasan oleh pihak *Security* agar tidak ada kebocoran data dari dalam kantor, dan ketika dicoba dengan koneksi *mobile data*, permasalahan ini selesai.

6. Menu Error : Menu *Button* (Gambar 2.34)

Deskripsi : Warna menu button suka bug, ketika saya klik *account* warnanya masih berada di *Home*.

Penyelesaian: Kami selesaikan dengan mengubah kode ketika halaman *account* pada *tab* di klik, sebelumnya kami melakukan *push* halaman *account* baru ke antar muka yang aktif, kemudian kami ubah menjadi melakukan *push* pada halaman *account* yang sudah disimpan sebelumnya di dalam *tab*.

7. Menu Error : Menu *back* (Gambar 2.35)

Deskripsi : Ketika kita di menu FAQ's, Lapor, *Inbox* dan *Account*, alangkah baiknya diberikan *button back*, karena disini cuma "detail blog dan detail *ticket* saja yang diberikan menu *back*".

Penyelesaian: Masalah ini diselesaikan dengan alasan bahwa halaman yang QC sebutkan adalah halaman yang ada di *tab* pada *bottom screen*, sedangkan halaman yang kami berikan tombol kembali hanyalah halaman yang kami *push*, bukan halaman yang ada di *root*.

8. Menu Error : Menu Lapor (Gambar 2.36)

Deskripsi : "Kronologi" dengan "garis untuk mengisi" terlalu jauh, alangkah lebih baik sama seperti diatasnya tetapi bisa melebar ketika karakter lebih dari *text box*.

Penyelesaian: Permasalahan ini diselesaikan oleh teman saya Budi, dengan membuat kolom input menjadi responsif berdasarkan isi teks yang telah di inputkan, apabila kolom *default* tidak cukup maka akan secara otomatis melebar seiring bertambahnya karakter yang pengguna ketikan.

9. Menu Error : *Category* Pelanggaran Etika Kerja (Gambar 2.37)

Deskripsi : Keterangan PIC Tidak keluar

Penyelesaian: Saya selesaikan dengan mengubah kode pada saat membuat tiket baru, karena sebelumnya saya tidak tau kode PIC didapat dari mana, maka saya membuat agar secara default kolom PIC dikosongkan.

10. Menu Error : Menu Lapor (Gambar 2.38)

Deskripsi : Setelah *Fixing* aplikasi, menu lapor tidak bisa di simpan, sudah di coba ganti kategori dan *sub* kategorinya tidak bisa juga.

Penyelesaian: Hal ini terjadi karena *captcha* yang telah saya buat menghilang setelah 1 kali membuat laporan, sehingga tombol *submit* yang memerlukan *captcha* agar selesai, namun *captcha* tidak muncul untuk dapat diselesaikan. Masalah ini saya selesaikan dengan me-reload *captcha* yang aktif, karena sebelumnya sekalipun setiap kolom *input* di *reset* setelah selesai melakukan pelaporan, *captcha* tidak di *reset*.

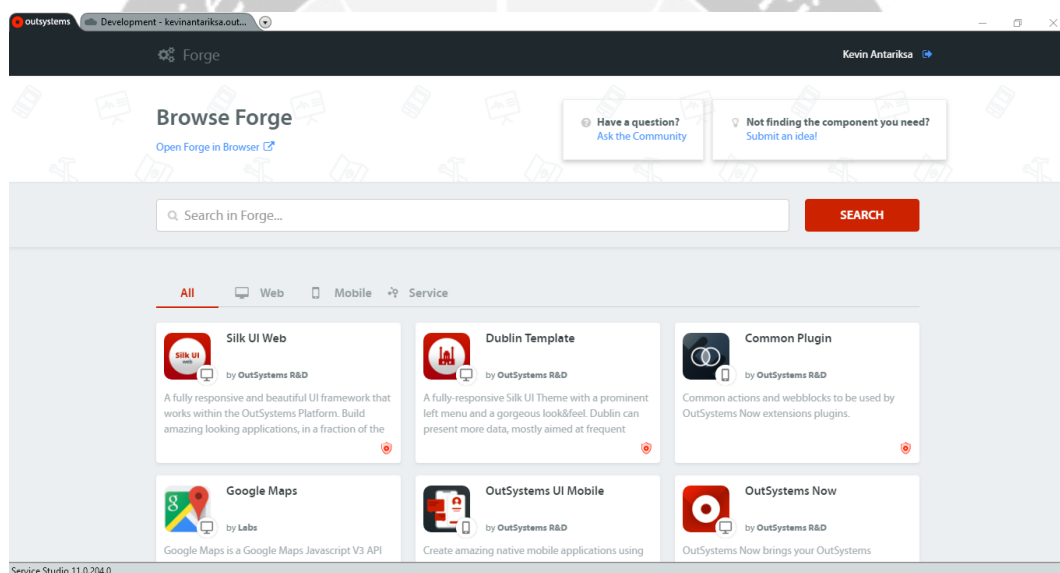
Senin, 1 April 2019 - Jumat, 5 April 2019

Project kami telah sampai pada tahap akhir, 1 minggu ini mas Ardy meminta Budi untuk *develop* aplikasi ini dengan *remote* Mac di kantor Jakarta agar dapat di *compile* di Mac sehingga dapat di *install* pada iOS dan saya selama seminggu ini bertugas sebagai *support* apabila Budi memerlukan bantuan.

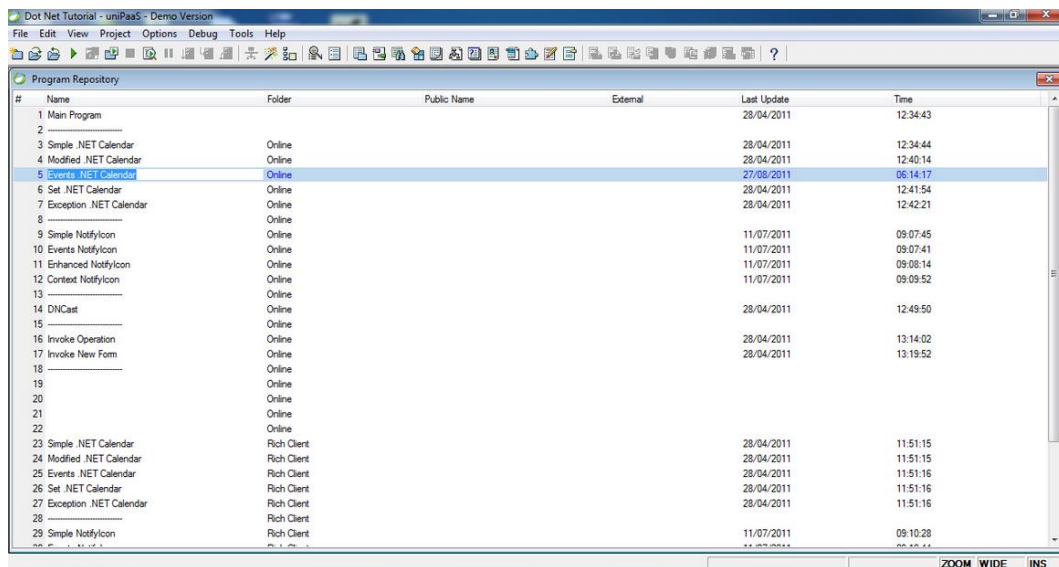
2.2 Hasil Pekerjaan Secara Umum

Berdasarkan magang yang telah saya lakukan, hasil yang saya kerjakan adalah sebuah aplikasi bernama AccWhistle yang berguna untuk pelaporan atas keluhan *customer* dari perusahaan Astra Credit Company. Dalam proses bisnisnya, perusahaan ACC menerima segala kritik dan saran dari *customer*. Oleh karena itu dibuatlah aplikasi bernama AccWhistle ini. Aplikasi ini merupakan aplikasi yang berbasis Android, iOS dan Website yang memiliki *database* yang sama.

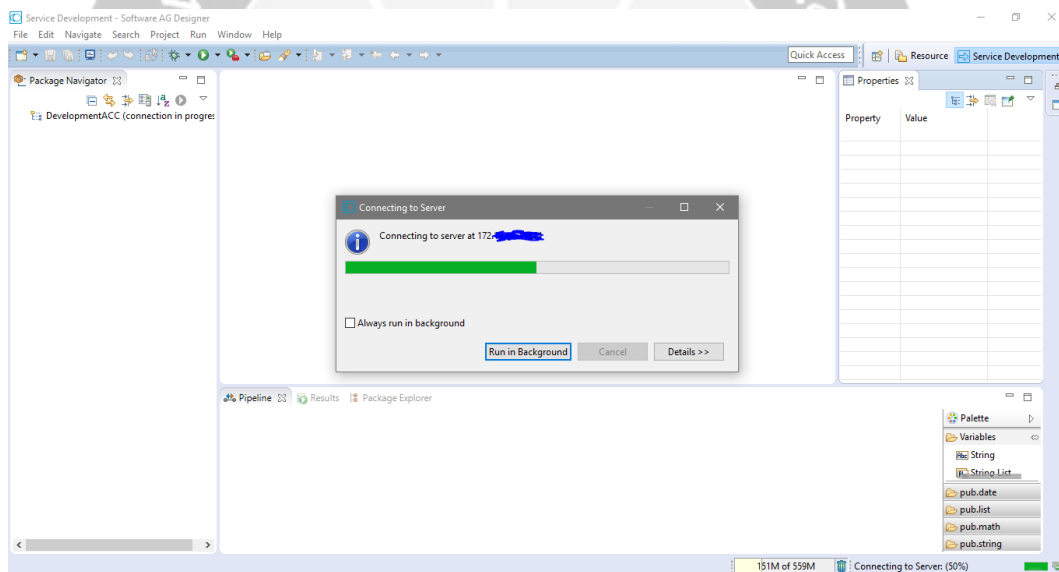
2.3 Bukti Hasil Pekerjaan



Gambar 2.1 Aplikasi OutSystem



Gambar 2.2 Aplikasi uniPaaS

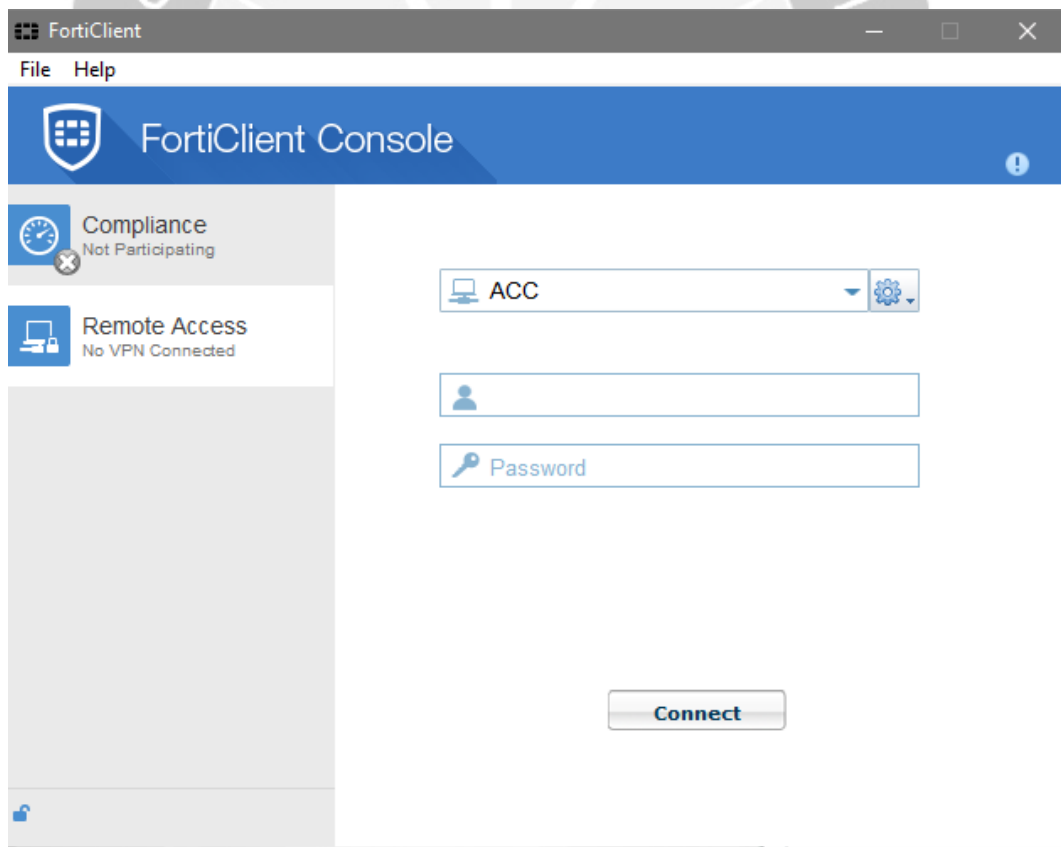


Gambar 2.3 Aplikasi Software AG Designer

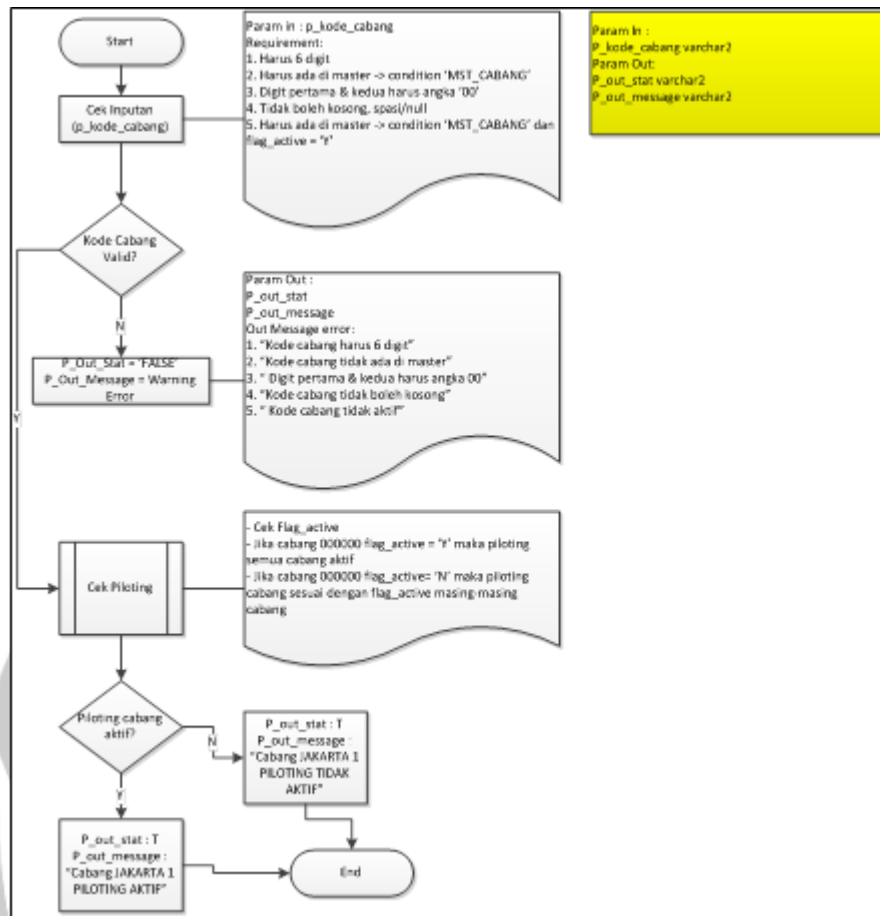
Laravel

[DOCUMENTATION](#) [LARACASTS](#) [NEWS](#) [FORGE](#) [GITHUB](#)

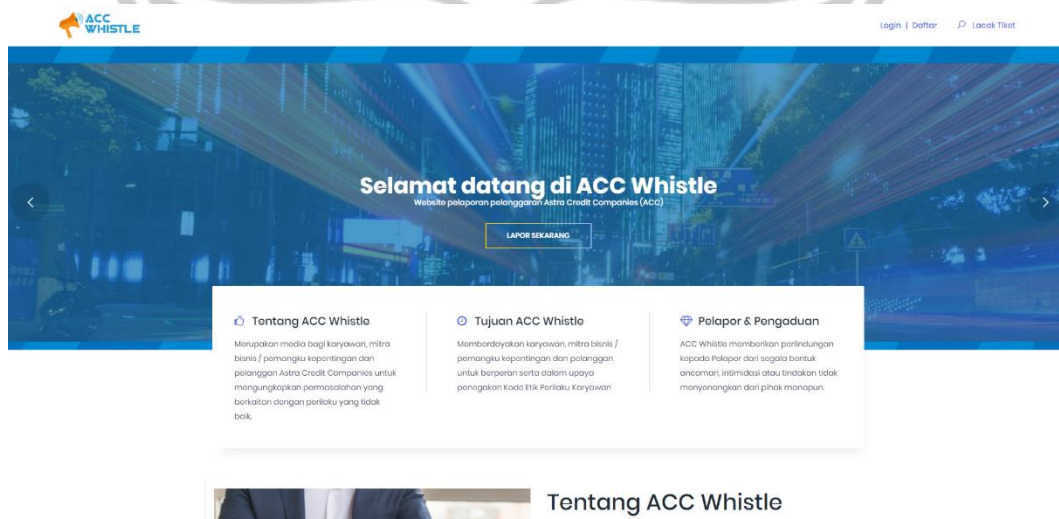
Gambar 2.4 Welcome Page Laravel



Gambar 2.5 FortiClient VPN



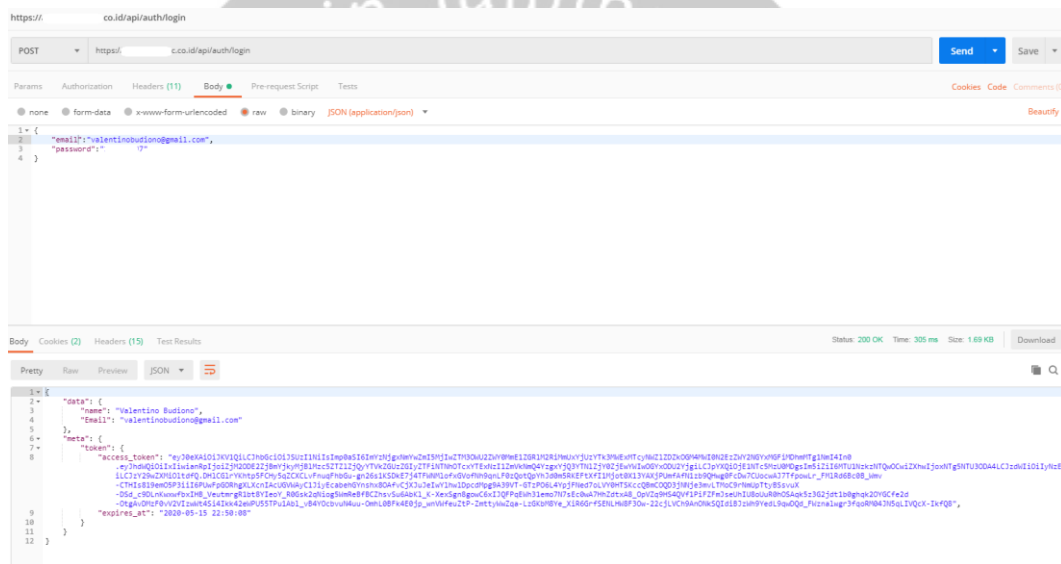
Gambar 2.6 Flowchart PL/SQL yang dibuat



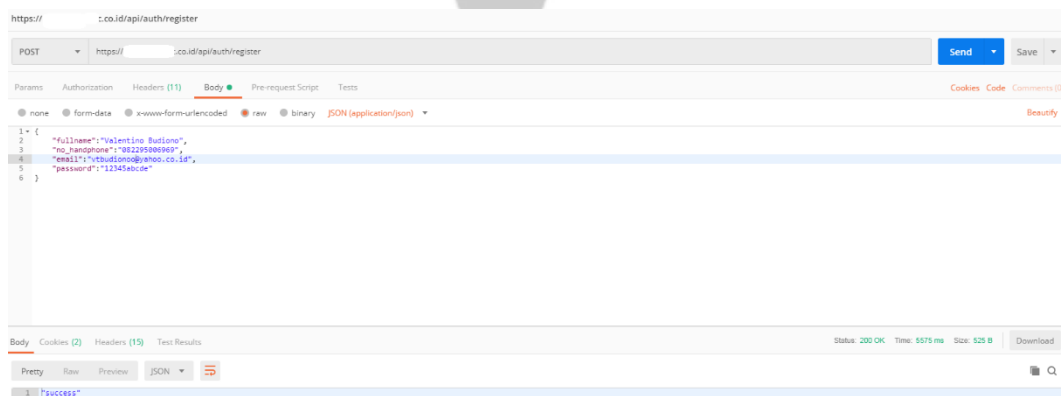
Gambar 2.7 Website AccWhistle



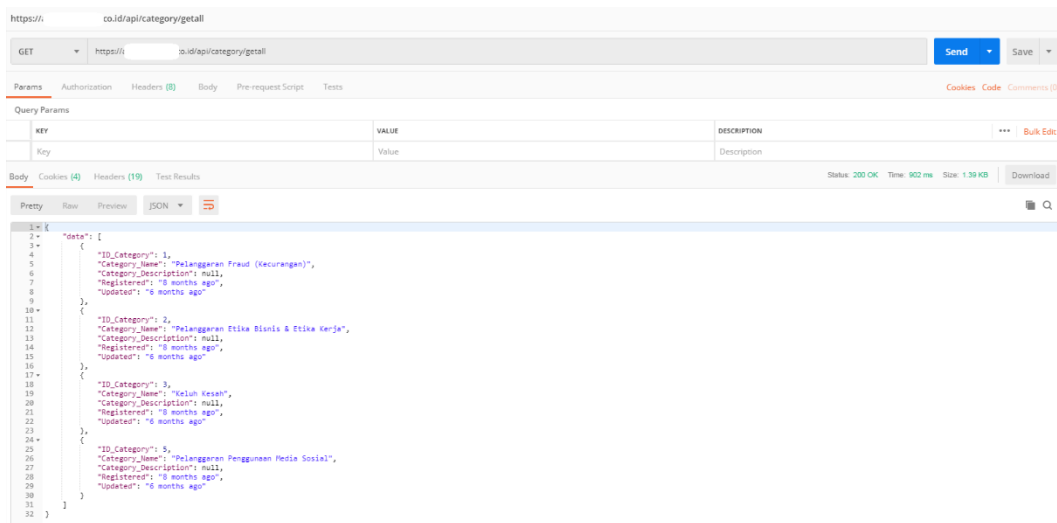
Gambar 2.8 Logo Ionic



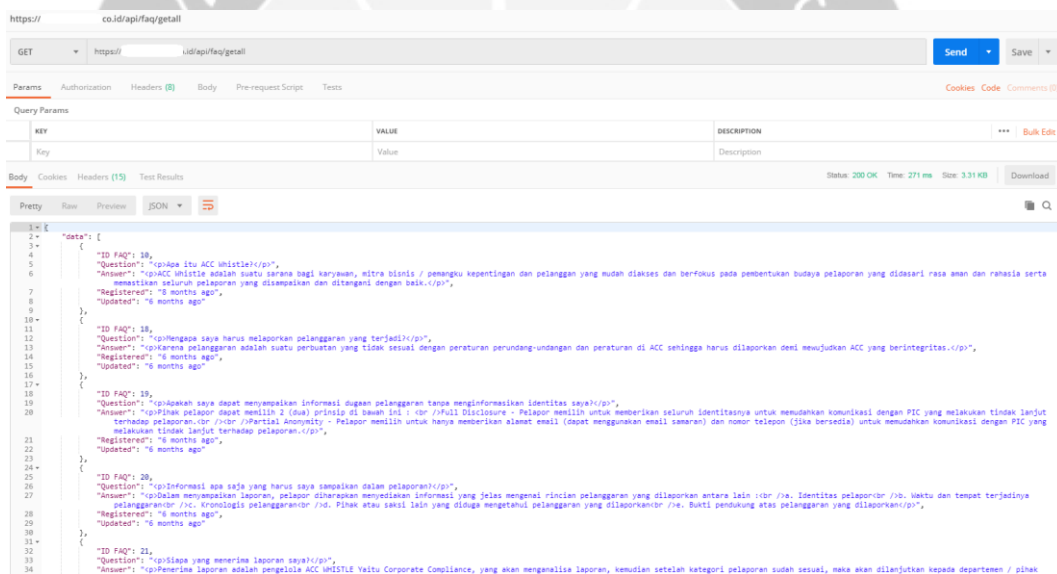
Gambar 2.9 API Login



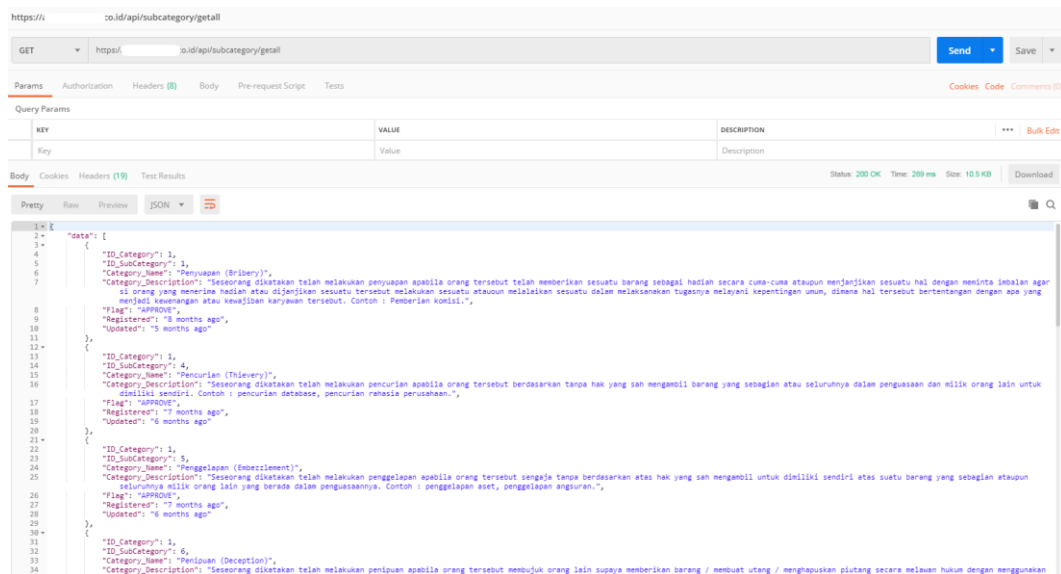
Gambar 2.10 API Register



Gambar 2.11 API getAllCategory



Gambar 2.12 API getAllFAQ



Gambar 2.13 API getAllSubCategory

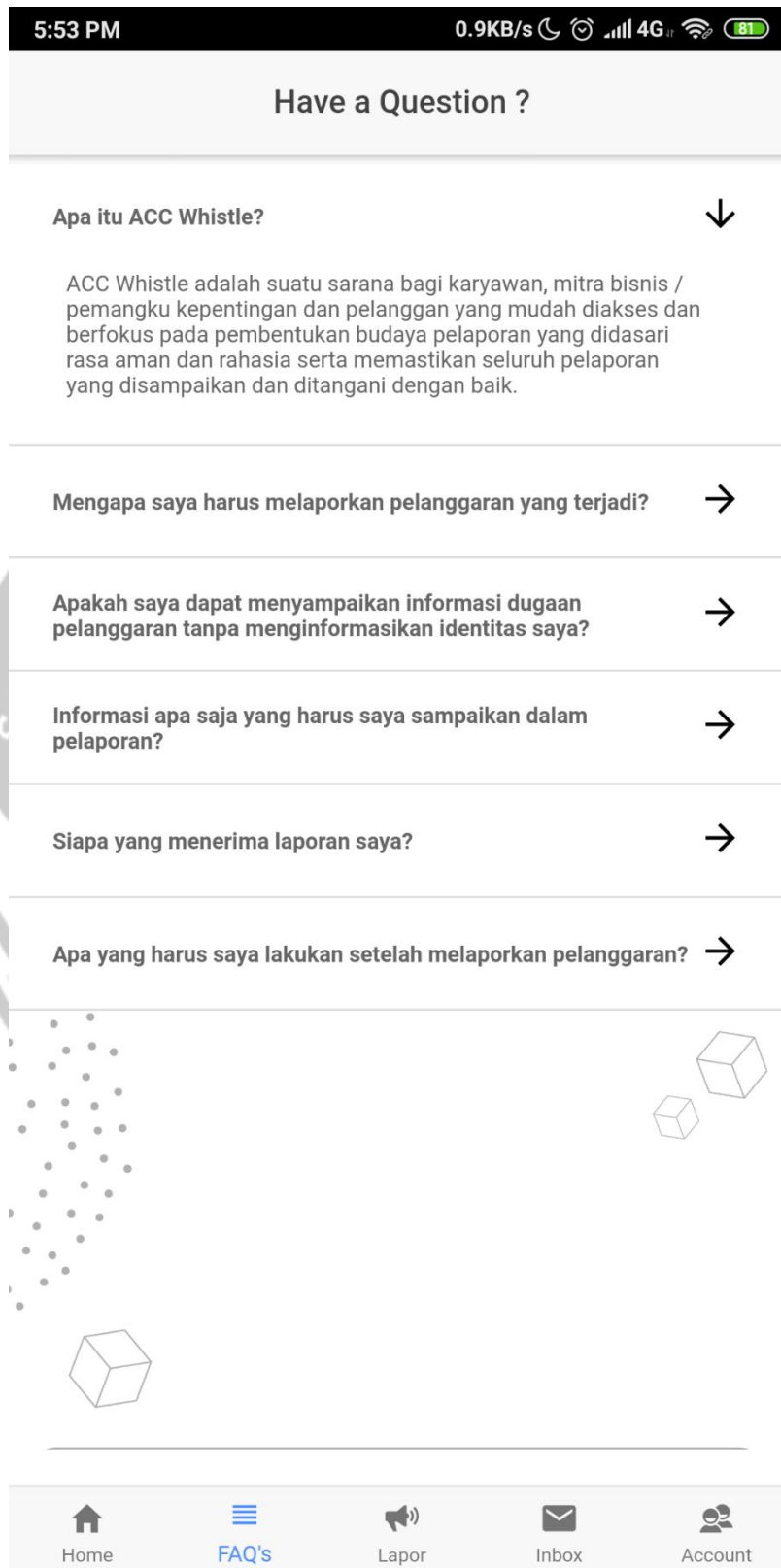
```

public function getMyTicketList(Request $request)
{
    // $user = User::where('id',$request->id_user)->first();
    $user = $request->user();

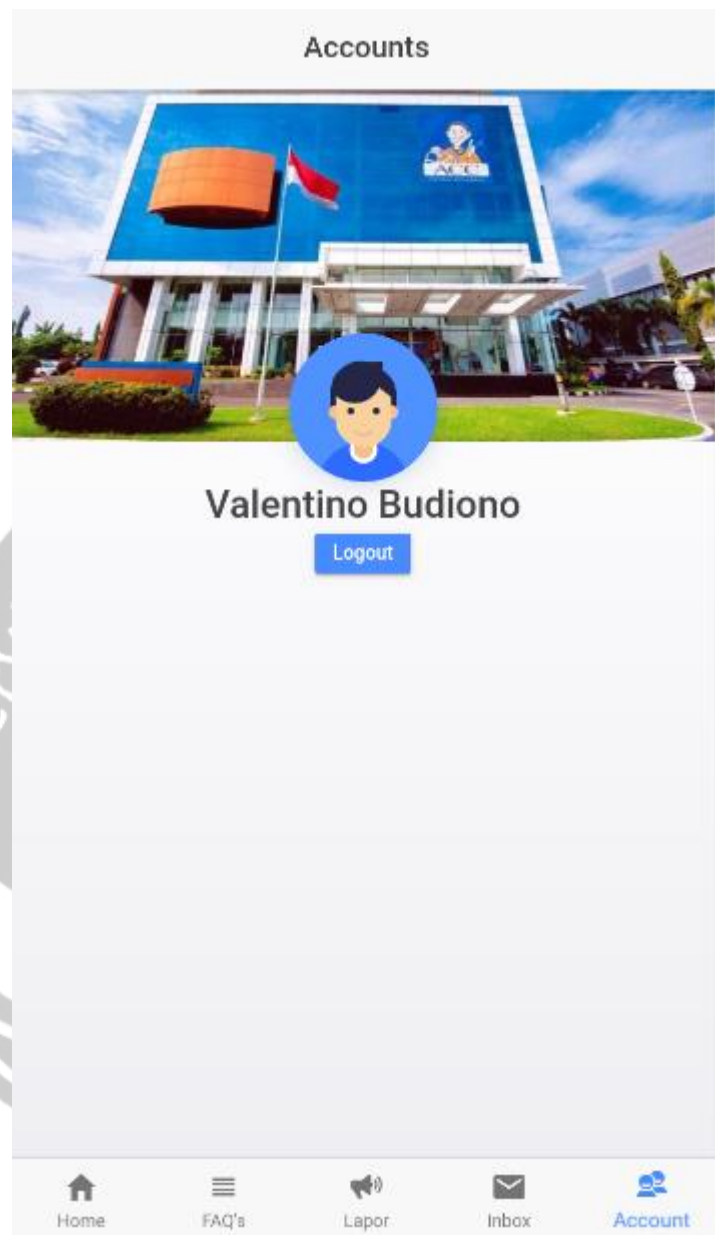
    if (empty($user)) {
        $tickets = $user->tickets()->get();
        for ($i = 0; $i < $tickets->count(); $i++) {
            $data[$i]['id'] = $tickets[$i]['id_ticket'];
            $data[$i]['kategori'] = ParentCategory::where('id', $tickets[$i]['id_category'])->get()->first()->category_name;
            $data[$i]['sub_kategori'] = SubCategory::where('id_category', $tickets[$i]['id_category'])->where('id_sub_category', $tickets[$i]['id_sub_category'])->get()->first()->sub_category_name;
            $data[$i]['status'] = $tickets[$i]['ticket_status'];
            $data[$i]['sla'] = $tickets[$i]['ticket_sla'];
            $data[$i]['flag_read'] = $tickets[$i]['flag_read'];
            $data[$i]['created_at'] = $tickets[$i]['created_at']->format('Y-m-d H:i:s');
            $data[$i]['updated_at'] = $tickets[$i]['updated_at']->format('Y-m-d H:i:s');
        }
        return response()->json($data);
    } else {
        return response()->json([
            'message' => 'user not found',
            'status' => 'failed',
        ], 204);
    }
}

```

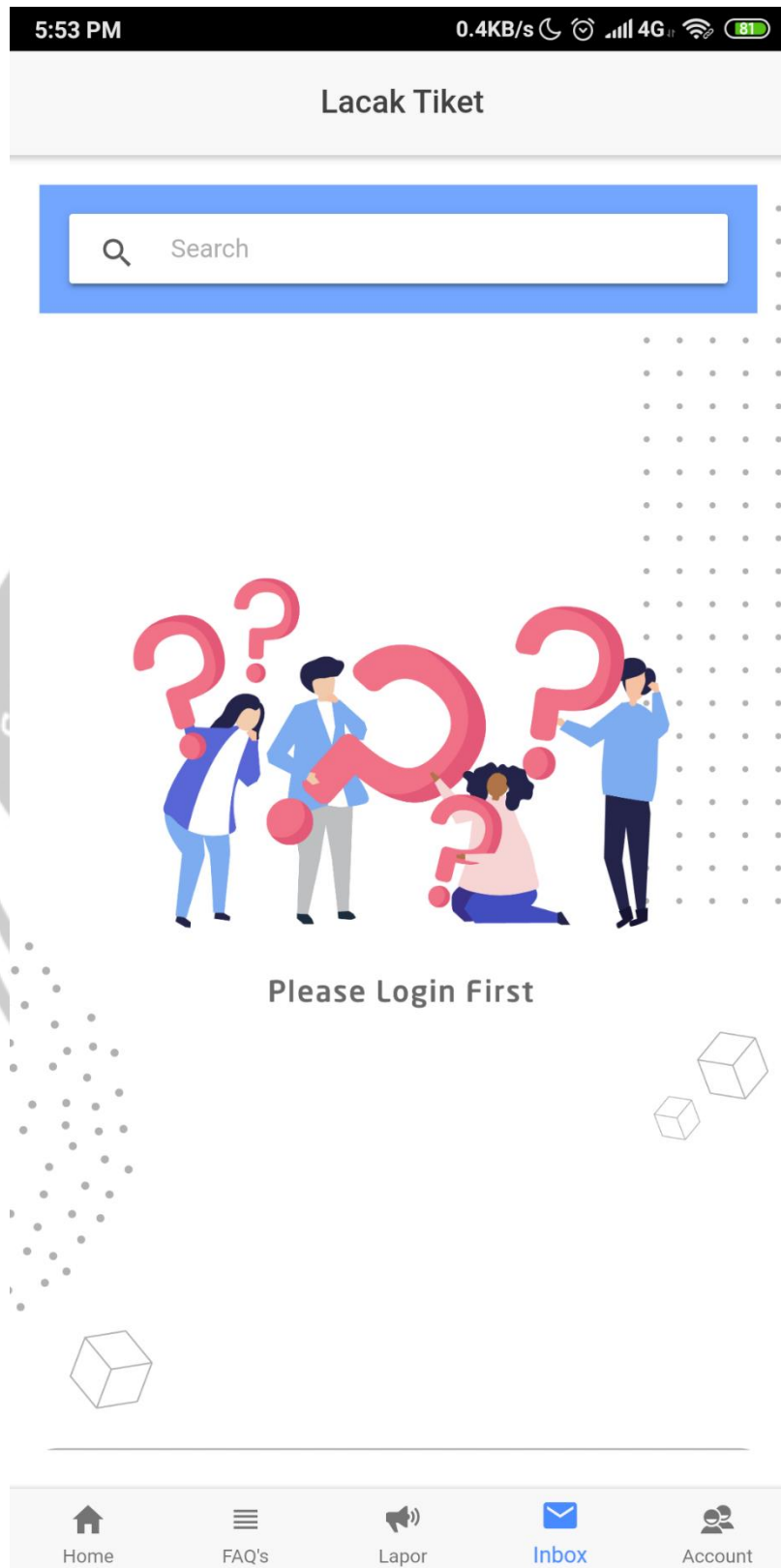
Gambar 2.14 Fungsi getMyTicketList



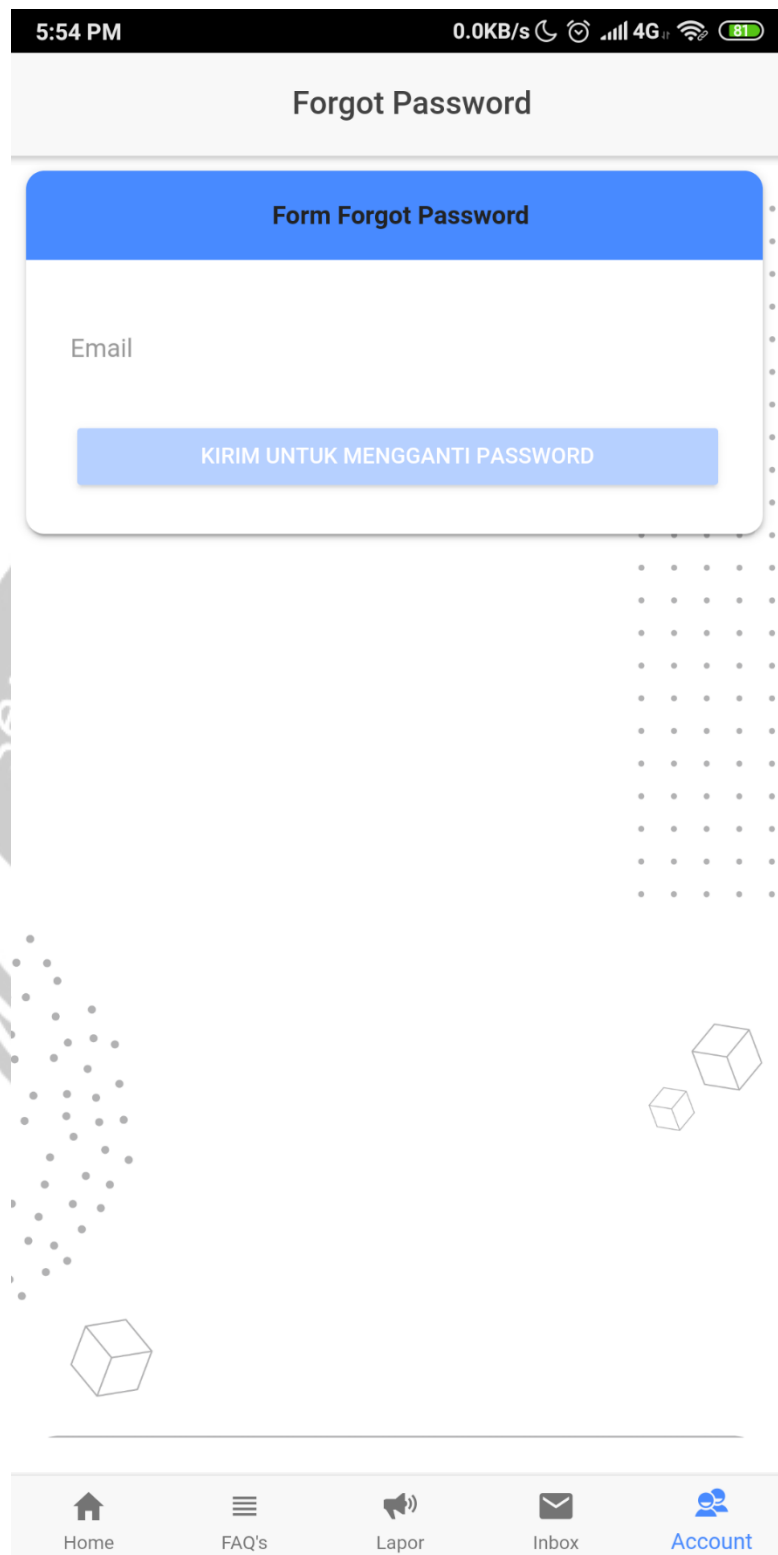
Gambar 2.15 Halaman FAQ pada Aplikasi Android



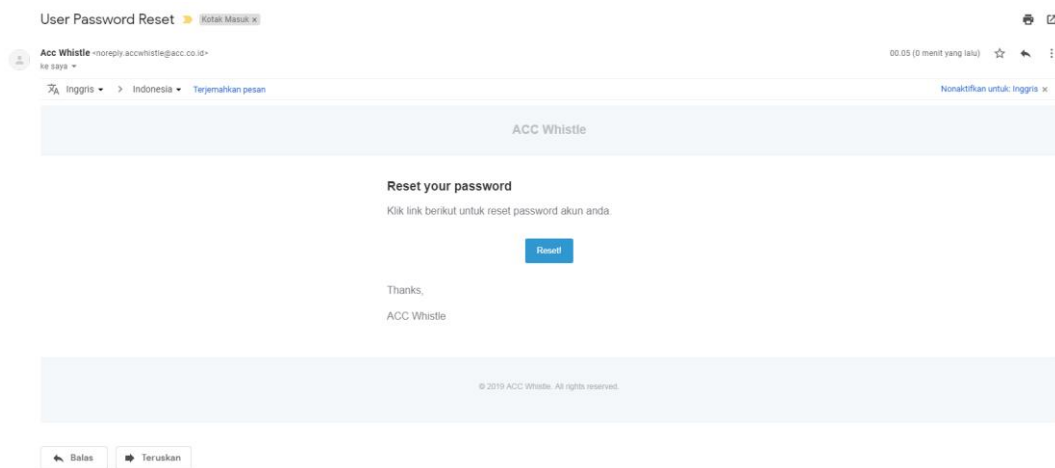
Gambar 2.16 Halaman Profile setelah Login



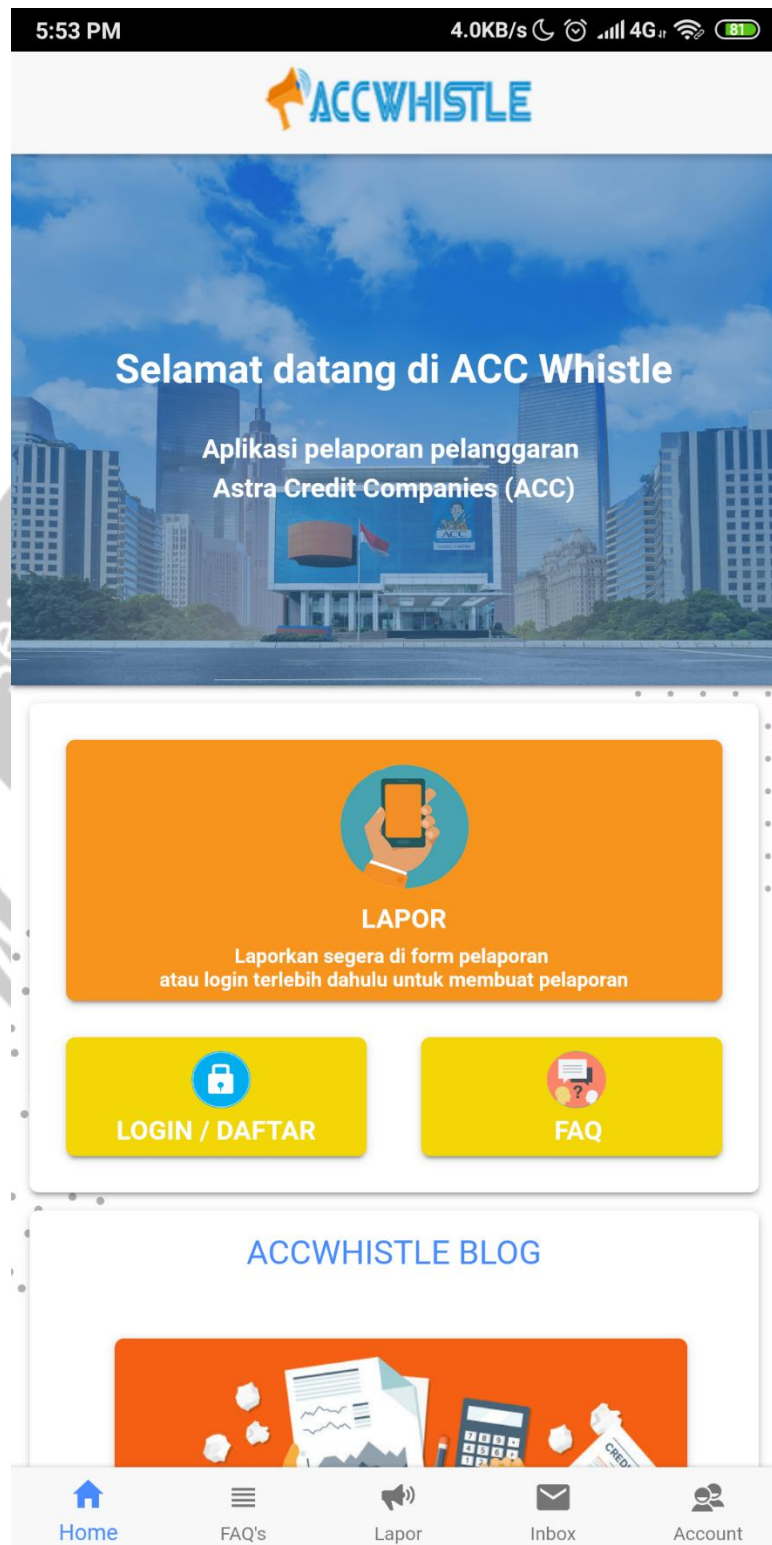
Gambar 2.17 Halaman Lacak Tiket



Gambar 2.18 Halaman Reset Password



Gambar 2.19 Email untuk Reset Password



Gambar 2.20 Halaman Home

```

<ion-content class="card-background-page">
  <ion-grid style="padding: 0;">
    <ion-row>
      <ion-col style="padding: 0;">
        <ion-card style="max-height: 100%; width:100%; margin-left: 0; border-radius: 0; margin-top: 0%; float: left;">
          <ion-card-content text-center style="padding: 0; padding-bottom: 0;">
            <ion-grid style="padding: 0; padding-bottom: 0;">
              <ion-row>
                <ion-col style="padding: 0; padding-bottom: 0;">
                  <ion-card class="card-welcome">
                    
                    <ion-label class="card-title">Selamat datang di ACC Whistle</ion-label>
                    <ion-label class="card-title-2" text-wrap>Aplikasi pelaporan pelanggaran <br>Astra Credit Companies (ACC)</ion-label>
                  </ion-card>
                </ion-col>
              </ion-row>
            </ion-grid>
          </ion-card-content>
        </ion-card>
      </ion-col>
    </ion-row>
  </ion-grid>
</ion-content>

```

Gambar 2.21 Tag Khusus dari Ionic

```

this.file
  .resolveLocalFileSystemUrl(resolvedFilePath)
  .then(fileInfo => {
    this.count[inde + 1] = "buttonFileCounter";
    fileInfo.getMetadata(metadata => {
      if (
        this.sourceFileType[inde] == "JPG" ||
        this.sourceFileType[inde] == "jpg" ||
        this.sourceFileType[inde] == "gif" ||
        this.sourceFileType[inde] == "doc" ||
        this.sourceFileType[inde] == "pdf" ||
        this.sourceFileType[inde] == "docx" ||
        this.sourceFileType[inde] == "xls" ||
        this.sourceFileType[inde] == "xlsx" ||
        this.sourceFileType[inde] == "ppt" ||
        this.sourceFileType[inde] == "pptx" ||
        this.sourceFileType[inde] == "jpeg" ||
        this.sourceFileType[inde] == "png" ||
        this.sourceFileType[inde] == "mpga" ||
        this.sourceFileType[inde] == "mp3" ||
        this.sourceFileType[inde] == "mpeg" ||
        this.sourceFileType[inde] == "mp4" ||
        this.sourceFileType[inde] == "3gp"
      ) {

```

Gambar 2.22 Kode untuk filterisasi tipe file

LAPOR

Form Pelaporan

Anda tidak perlu khawatir terungkapnya identitas diri anda karena ACC Whistle akan **MERAHSAKANTAS IDNTITAS DIRI ANDA** sebagai whistleblower. ACC Whistle menghargai informasi yang Anda laporkan. Fokus kami hanya kepada materi informasi yang Anda Laporkan.

Email *
Silahkan masukan email anda.

valentinobudiono@gmail.com

Pilih Kategori *
Silahkan pilih kategori pelaporan.

Detail Kategori *
Silahkan pilih detail kategori pelaporan.

Uraikan kasus dugaan pelanggaran secara rinci. Apa dugaan pelanggaran yang anda ketahui, Dimana perbuatan pelanggaran tersebut dilakukan, Kapan perbuatan pelanggaran tersebut dilakukan. Siapa saja yang terlibat dalam perbuatan pelanggaran, bagaimana perbuatan pelanggaran tersebut dilakukan. Mohon untuk dibold seperti contoh terlampir diatas.

Kronologi *
Jelaskan kronologi

Pilih File
Anda dapat melampirkan file data/dokumen pendukung. Maksimum ukuran file 10 MB per input file upload.
type file =jpg,gif,doc,pdf,docx,xls, xlsx,ppt,ppix,jpeg,ong,mpgs,mp3,mpeg,mp4,3gp.

Choose file

SIMPAN

BATAL

Home

FAQ's

Lapor

Inbox

Account

Gambar 2.23 Halaman Lapor setelah Login dengan auto fill email

5:53 PM
8.0KB/s
4G
81

LAPOR

Form Pelaporan

Anda tidak perlu khawatir terungkapnya identitas diri anda karena ACC Whistle akan MERAHASIAKAN IDENTITAS DIRI ANDA sebagai whistleblower. ACC Whistle menghargai informasi yang Anda laporkan. Fokus kami hanya kepada materi informasi yang Anda Laporkan.

Email *
Silahkan masukan email anda.

No handphone (Jika Bersedia)

Pilih Kategori *
Silahkan pilih kategori pelaporan.

Detail Kategori *
Silahkan pilih detail kategori pelaporan.

Uraikan kasus dugaan pelanggaran secara rinci. Apa dugaan pelanggaran yang anda ketahui, Dimana perbuatan pelanggaran tersebut dilakukan, Kapan perbuatan pelanggaran tersebut dilakukan. Siapa saja yang terlibat dalam perbuatan pelanggaran, bagaimana perbuatan pelanggaran tersebut dilakukan. Mohon untuk dibold seperti contoh terlampir diatas.

Kronologi *
Jelaskan kronologi

Pilih File
Anda dapat melampirkan file data/dokumen pendukung. Maksimum ukuran file 10 MB per input file upload.
type file =jpg,gif,doc,pdf,docx,xls, xlsx,ppt,pptx,jpeg,png,mpga,mp3,mpeg,mp4,3gp.

Choose file

☐ I'm not a robot

reCAPTCHA

Home
FAQ's
Lapor
Inbox
Account

Gambar 2.24 Halaman Lapor saat Pengguna belum Login

```

if (
  metadata.size > 10000000 ||
  this.sourceFileType[inde] == ""
) {

```

Gambar 2.25 Kode untuk filterisasi file diatas 10 MB

```

<div *ngFor="let i of count; let x = index">
  <ion-grid>
    <ion-row>
      <ion-col style="align-self: center;" align-self-left col-4>
        <button (click)="chooseFile(x)">Choose file</button>
      </ion-col>
      <ion-col style="text-align: left;" align-self-left col-8>
        <ion-label style="margin: 0; padding-left: 20%;" *ngIf="count[x+1]">
          {{sourceFileName[x] | truncate:[20]}}</ion-label>
        </ion-col>
      </ion-row>
    </ion-grid>
  </div>

```

Gambar 2.26 Kode ngFor dari AngularJS pada file HTML

ACCWHISTLE MOBILE SIT LIST ERROR

No	Menu Error	Description	Screen Menu	Status	QC	Developer	Keterangan Status
1	1 Input No Handphone	Pada input no masih bisa menggunakan huruf, harusnya angka saja					
2	2 Administrator Change Email User	Ketika admin ganti email user tidak ada konfirmasi verifikasi ulang ke user dimana berdasarkan pada user yang tidak mengetahui perubahan tersebut tidak dapat login/lapor dan bersangkutan juga kepada laporan tidak akan masuk ke email user.		Close	Caesar		Selesai
3	3 Menu Account	Request avatar di account bisa pakai foto, karena di menu account sudah simple		Request Close	Caesar		User tidak minta foto, hanya sebagai icon saja.
4	4 Menu Login	pada menu login ada keterangan "belum punya akun?" dan "daftar disini" lebih baik di gabung karena jika tidak tulisan "belum punya akun" akan rancu karena tidak ada perbedaan yang bisa di klik dan tidak.		Request	Caesar		
5	5 Loading	Loading logout suka lama		Close	Caesar		Selesai
6	6 Menu Button	Warna menu button suka bug, ketika saya klik account warnanya malah berada di Home		Close	Caesar		Selesai
7	7 Menu back	Ketika kita di menu FAQ's, Lapor, Inbox dan Account, alangkah baiknya diberikan button back, karena disini cuma "detail blog dan detail ticket saja yang diberikan menu back".		Request	Caesar		

Gambar 2.27 Bug yang ditemukan dilist pada google spreadsheets

← Pendaftaran

Form Pendaftaran

Nama Lengkap

Muh caesar

Email

Jend.caesar@gmail.com

No Handphone

08satuduatiga

The no handphone format is invalid!

Password

Konfirmasi Password

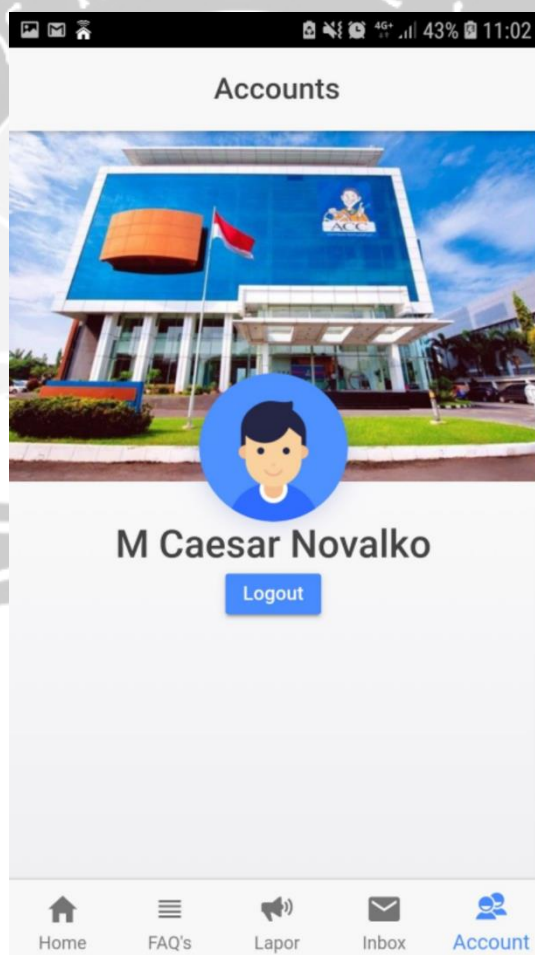
DAFTAR BATAL

Home FAQ's Lapor Inbox Account

Gambar 2.28 Bug Pertama



Gambar 2.29 Bug Kedua




Gambar 2.30 Request QC Pertama

← LOGIN

Form Login

Pelapor memilih untuk memberikan seluruh identitasnya untuk memudahkan komunikasi dengan PIC yang melakukan tindak lanjut terhadap pelaporan. Dengan memilih pelaporan Full Disclosure, anda dapat menggunakan fitur lacak tiket untuk memantau pelaporan anda.

Username

Password 

[LOGIN](#) [BATAL](#)

Belum punya akun?

[Daftar disini!](#)

[Lupa Password ?](#)

Home FAQ's Lapor Inbox Account

Gambar 2.31 Request QC Kedua

5:53 PM0.2KB/s🌙🕒📶4G📶81%

←

LOGIN

Form Login

Pelapor memilih untuk memberikan seluruh identitasnya untuk memudahkan komunikasi dengan PIC yang melakukan tindak lanjut terhadap pelaporan. Dengan memilih pelaporan Full Disclosure, anda dapat menggunakan fitur lacak tiket untuk memantau pelaporan anda.

Username

Password

👁

LOGIN

BATAL

Belum punya akun? [Daftar disini!](#)

Lupa Password?

🏠Home

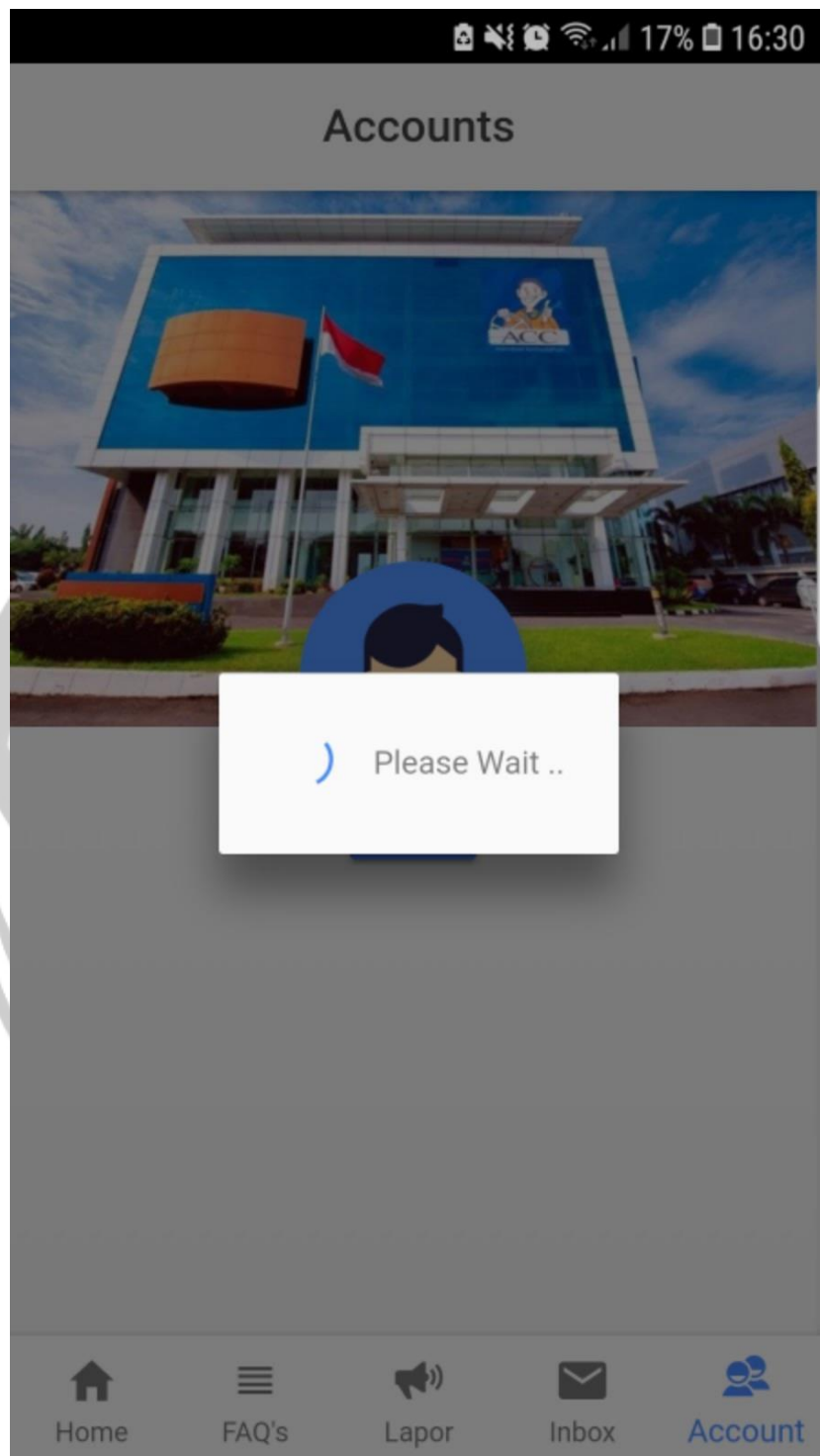
☰FAQ's

📢Lapor

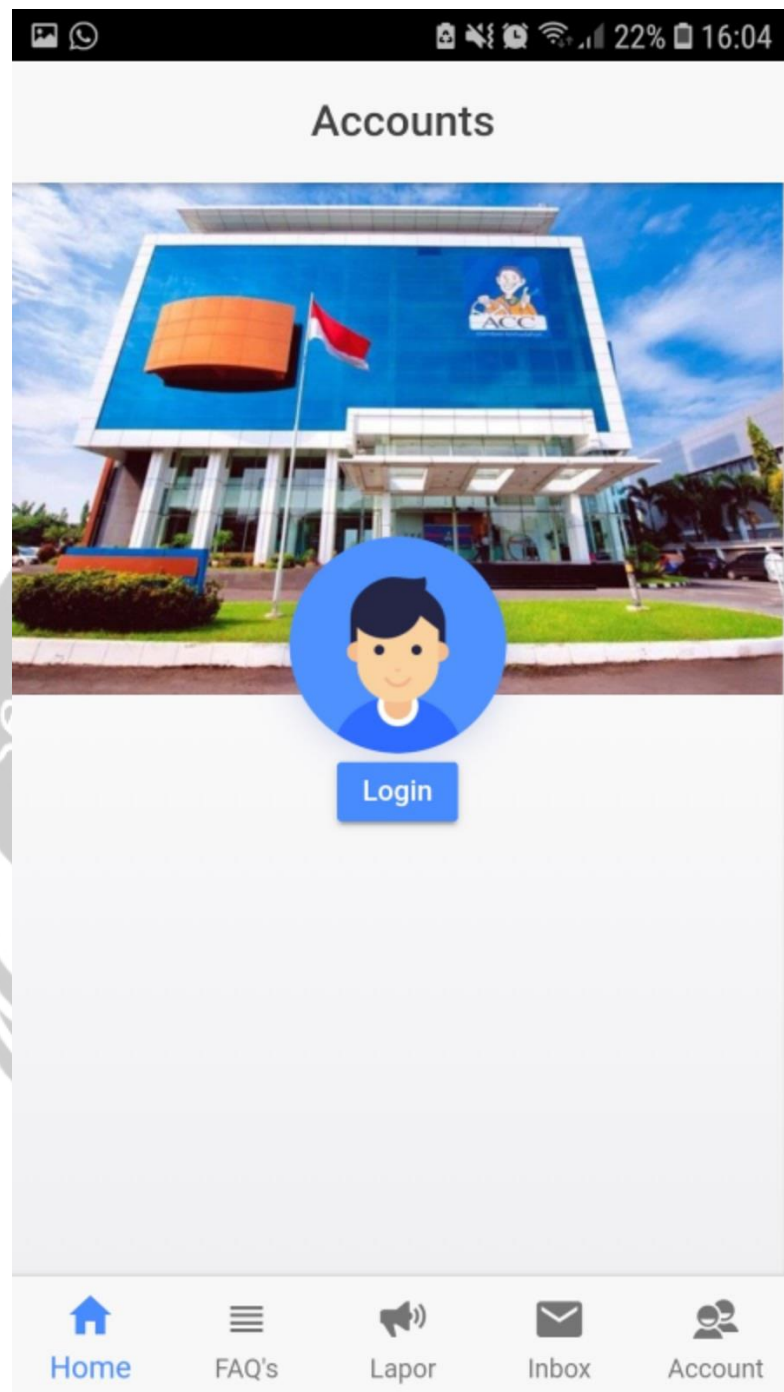
✉Inbox

👤Account

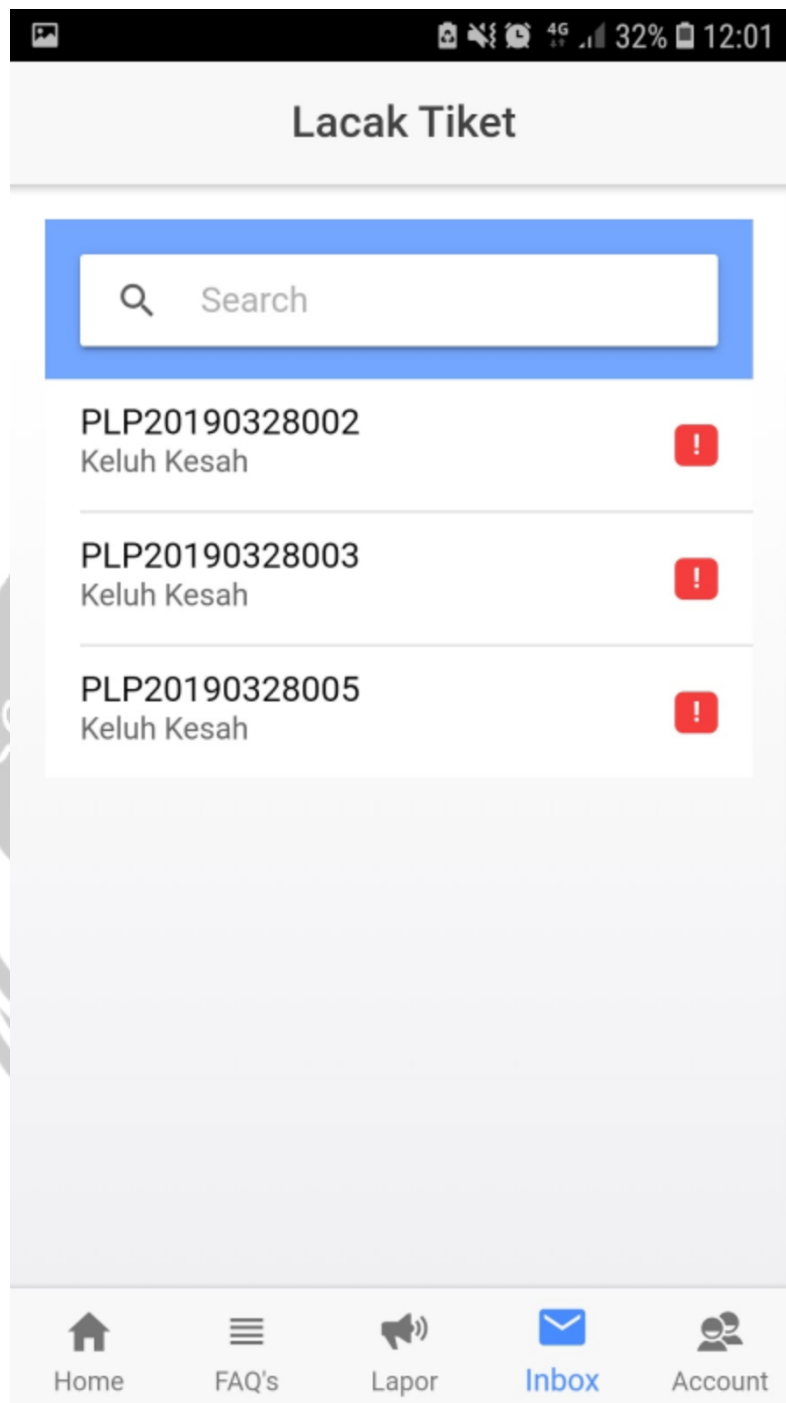
Gambar 2.32 Login Screen Baru



Gambar 2.33 Bug Ketiga



Gambar 2.34 Bug Keempat



Gambar 2.35 Request QC Ketiga

LAPOR

Uraikan kasus dugaan pelanggaran secara rinci.
 Apa dugaan pelanggaran yang anda ketahui,
 Dimana perbuatan pelanggaran tersebut dilakukan,
 Kapan perbuatan pelanggaran tersebut dilakukan.
 Siapa saja yang terlibat dalam perbuatan
 pelanggaran, bagaimana perbuatan pelanggaran
 tersebut dilakukan. Mohon untuk dibold seperti
 contoh terlampir diatas.

Kronologi *
 Jelaskan kronologi

Please fill out Sub Category field!

Pilih File

Anda dapat melampirkan file data/dokumen pendukung. Maksimum ukuran file 10 MB per input file

upload
 type file =jpg,gif,doc,pdf,docx,xls,
 xlsx,ppt,pptx,jpeg,png,mpga,mp3,mpeg,mp4,3gp.

Choose file

Home
FAQ's
Lapor
Inbox
Account

Gambar 2.36 Request QC Keempat

Show 10 entries

Search:

Ticket ID	Request Date	Last Response	Category	Sub Category	PIC	SLA	Status	Action
PUP20190329003	29 Mar 2019 10:30	29 Mar 2019 10:30	Pelanggaran Etika Bisnis & Etika Kerja	Konflik Kepentingan			RECS	

Gambar 2.37 Bug Kelima

LAPOR

Form Pelaporan

Anda tidak perlu khawatir terungkapnya identitas diri anda karena ACC Whistle akan MERAHASIAKAN IDENTITAS DIRI ANDA sebagai whistleblower. ACC Whistle menghargai informasi yang Anda laporkan. Fokus kami hanya kepada materi informasi yang Anda Laporkan.

Email *
Silahkan masukan email anda.

Mcaesarnovalko@gmail.com

Pilih Kategori *
Silahkan pilih kategori pelaporan.

Pelanggaran Fraud (Kecurangan) ▼

Detail Kategori *
Silahkan pilih detail kategori pelaporan.

Penyuapan (Bribery) ▼

Uraikan kasus dugaan pelanggaran secara rinci. Apa dugaan pelanggaran yang anda ketahui, Dimana perbuatan pelanggaran tersebut dilakukan, Kapan perbuatan pelanggaran tersebut dilakukan. Siapa saja yang terlibat dalam perbuatan pelanggaran, bagaimana perbuatan pelanggaran tersebut dilakukan. Mohon untuk dibold seperti contoh terlampir diatas.

Kronologi *

Testing IT QC

Pilih File

Anda dapat melampirkan file data/dokumen pendukung. Maksimum ukuran file 10 MB per input file

upload type file =jpg,gif,doc,pdf,docx,xls, xlsx,ppt,pptx,jpeg,png,mpeg,mp3,mp4,3gp.

Choose file

Screenshot_20190329...


Choose file


Screenshot_20190329...


Choose file


SIMPAN


BATAL


Home


FAQ's


Lapor


Inbox


Account

Gambar 2.38 Bug Keenam

BAB III

HASIL PEMBELAJARAN

3.1 Manfaat Magang

Magang memberikan sangat banyak manfaat berharga. Pertama, saya dapat merasakan suasana atau lingkungan kerja sesungguhnya sebagai seorang *developer*. Mengetahui lingkungan kerja dapat membuat saya menjadi lebih nyaman ketika benar-benar terjun ke dunia profesional nantinya.

Dengan melaksanakan magang, saya mengetahui bagaimana mengerjakan sebuah proyek yang digunakan pada sebuah perusahaan. Saya lebih memahami manfaat dari koordinasi dan kolaborasi dalam pengerjaan sebuah proyek. Meskipun dalam pengembangan aplikasi ini saya menghadapi berbagai macam masalah dan harus dapat mengatasinya, namun semua penyelesaian itu tidak dapat terlepas dari bantuan rekan saya Budi dan bantuan dari Ibu Christie.

Karena kami bekerja secara *remote*, yang berarti ketika kami ada pertanyaan maupun saat ingin berkomunikasi kami harus menunggu BSA ACC di Jakarta meluangkan waktu untuk video call, terasa tidak mudah. Kegiatan *stand up meeting* yang diwakilkan dalam aplikasi Slack (menggunakan Geekbot), mengajarkan saya mengenai pentingnya kerja sama dan komunikasi. Komunikasi diperlukan untuk memastikan masing-masing anggota tim dapat mengerjakan tugasnya sesuai dengan yang diharapkan. Dengan adanya komunikasi yang baik, kendala yang dihadapi akan lebih mudah dicari penyelesaiannya.

Dalam pengerjaan proyek aplikasi AccWhistle ini, saya menyadari bahwa sebuah aplikasi yang telah selesai dibuat secara fungsionalitas masih dapat dikembangkan. Selain itu dengan dibuatnya API untuk setiap proses yang berhubungan dengan database dalam project AccWhistle ini akan dapat berguna lebih apabila API tersebut akan digunakan dalam aplikasi web/mobile lain. Penamaan *variable* dan penulisan *code* juga sangat penting karena bukan hanya kita yang akan menggunakan *code* itu. Pelaksanaan magang juga membuat saya

mengetahui tentang teknologi-teknologi yang saat ini digunakan di bidang industri IT, setidaknya pada lingkup perusahaan tempat saya magang.

3.2 Penerapan Ilmu dalam Magang

Pelaksanaan magang ini dapat dilakukan dengan baik tidak terlepas dari peranan ilmu pengetahuan dan pengalaman yang didapatkan dalam perkuliahan. Pertama adalah kerja sama dalam sebuah tim. Sebagai mahasiswa, kerja sama tim terbentuk dari pengerjaan tugas perkuliahan yang diberikan secara berkelompok maupun keikutsertaan dalam berbagai organisasi.

Mata kuliah Pemrograman Basis Data yang telah mengajari mengenai PL/SQL (*Procedural Language for SQL*). PL/SQL ini akan sangat berguna dalam pengembangan aplikasi-aplikasi berbasis Desktop, terutama apabila database bersifat *credential*. PL/SQL ini juga sangat mempermudah *developer* dalam mengembangkan aplikasinya karena tinggal memanggil nama prosedur beserta parameternya saja. Saat magang, saya hanya menggunakan PL/SQL ini pada masa training karena saya tidak mendapat proyek berbasis Desktop.

Mata kuliah Pengembangan Aplikasi Web (PAW) yang banyak mengajari mengenai *framework* Laravel. Selain itu mata kuliah ini mengajari bahasa dasar yang digunakan dalam pengembangan aplikasi web seperti HTML, JS, CSS, dan PHP. Seluruh elemen yang saya sebutkan tadi merupakan komponen kunci dalam magang saya. Sebelum mengikuti magang, saya tidak akan pernah tahu bahwa Ionic dan Laravel dapat dikombinasikan untuk membuat sebuah aplikasi yang sangat *powerful* dan pengguna *friendly*. Saya juga baru mengerti apabila Ionic dapat melakukan hal-hal ‘ajaib’ seperti pembuatan Pengguna Interface yang sangat bagus. Saya merasa akan sangat berguna apabila hal ini diajarkan dalam mata kuliah PAW.

BAB IV

KESIMPULAN

Magang yang saya lakukan pada tanggal 3 Januari – 7 April 2019, untuk perusahaan ACC (PT. Astra Sedaya Finance) telah berjalan dengan baik dan lancar berkat dukungan dari berbagai pihak. Dalam pelaksanaan magang ini saya satu tim dengan Michael Budi Santoso. Saya ditugaskan untuk mengerjakan proyek AccWhistle untuk aplikasi pelaporan oleh pelanggan. Proyek AccWhistle telah selesai kami kembangkan dengan memiliki menu untuk pelaporan atas ketidakpuasan pelanggan, melaporkan dan melacak laporan yang sudah di submit sebelumnya, artikel pada blog untuk pengguna baca dan pertanyaan umum yang biasanya ditanyakan oleh customer ACC terkait proses pelaporan atas ketidakpuasan. Dalam pengerjaan project ini saya mengalami beberapa kendala. Namun, dengan adanya peranan pembimbing lapangan dan rekan-rekan yang magang di ACC, kendala-kendala tersebut dapat diatasi dengan baik.

Pelaksanaan magang memberikan banyak manfaat bagi saya yang akan melanjutkan ke dunia kerja setelah menyelesaikan perkuliahan. Dengan melaksanakan magang, saya mengetahui bagaimana suasana dan lingkungan kerja dibidang teknologi, pengalaman mengerjakan proyek, serta pentingnya koordinasi, kerjasama, dan komunikasi. Selain itu, saya juga menyadari bahwa teknologi berkembang dengan sangat cepat. Oleh karena itu, diperlukan untuk selalu mempelajari teknologi-teknologi terbaru.

Pada pelaksanaan magang ini saya juga dapat menerapkan ilmu pengetahuan dan pengalaman yang saya dapatkan selama melaksanakan perkuliahan. Meskipun tidak seluruh ilmu dan pengetahuan yang dibutuhkan dalam magang ini diajarkan pada perkuliahan, namun dasar-dasar pengetahuan tersebut sebagian besar telah diajarkan. Saya juga menyadari bahwa saya harus selalu memperbarui pengetahuan, terbuka terhadap hal-hal baru, dan tidak hanya berpaku pada apa yang disampaikan pada bangku perkuliahan.